JOURNAL OF INEQUALITIES IN PURE AND APPLIED MATHEMATICS (JIPAM) WEB SITE DYNAMIC DATABASE SYSTEM

T. J. McGUCKIN

MASTER OF SCIENCE (4 UNIT THESIS)
2001

SCHOOL OF COMMUNICATIONS & INFOMATICS
VICTORIA UNIVERSITY OF TECHNOLOGY

Journal of Inequalities in Pure and Applied Mathematics

JIPAM WEB SITE

Dynamic Database System

by
Trevor McGuckin

A four (4) Unit Research Project/Thesis
for partial fulfillment of the degree
of Master of Science
(Mathematics & Computer Science)

January 2001
School of Communications and Infomatics
Faculty of Engineering and Science

DISCLAIMER

This thesis contains no material that has been accepted for the award of any other degree or diploma in any University or Tertiary Institute. To the best of my knowledge and belief it contains no material previously published or written by another person, except where reference is made in the text of the thesis.

SIGNATURE .....

NAME Trevor McGuckin

DATE 5/2/2001

ABSTRACT

The research project was framed around the need to convert a static page web site into a database driven web site using Microsoft technologies Active Server Pages and SQL server 7.

The methodology used for the software development was the object oriented use case approach of Jacobson.

Although a single software application was originally thought to be the best design strategy, the solution which emerged as being the preferred framework resulted in two software applications being developed.

Thus, the JIPAM Web Site Dynamic Database System is comprised of an application developed in ASP to allow online access to the journal and a second application to maintain the contents of the database which was developed in C++.

ACKNOWLEDGEMENTS

I would like to thank Steven Young, my supervisor, for his patience during the first semester when work commitments affected the amount of time I could find to work on this thesis.

I would also like to dearly thank my wife, Eileen Day. Firstly, I would like to thank her for her editorial effort in particular the work involved in making this thesis look very professional. Secondly, I would like to thank her for her involvement in testing the software towards the end of the development phase. Finally, I would like to thank her for her patience, support and encouragement with this undertaking.

TABLE OF CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

The School of Communication and Informatics within the Victoria University Faculty of Engineering and Science publish an academic journal titled, Journal of Inequalities in Pure and Applied Mathematics (JIPAM). In common with many academic other journals, both within the scientific community and also more broadly across numerous disciplines, the management board of JIPAM have chosen to utilize the opportunities provided by the Internet in terms of dissemination and delivery.

> *Scientific publishing on the World Wide Web makes it possible to disseminate new information to a wide audience around the world in a matter of minutes' (Weintraub, 2000, p.57).*
>
> *Electronic access and distribution of information have acquired a momentum which will continue to grow and thrive in the future with emerging technologies (Bandyopadhyay, 1999 p.14).*
>
> *Users, across the board, are becoming more savvy. Although the rise in online users varies widely across academic areas, they're all becoming increasingly demanding and knowledgeable (Fletcher, 1999, p.117).*

The first issue of JIPAM was produced in both online and the more traditional paper-based format. According to the Managing Editor, John Roumeliotis, 'E-publishing lets us compete with major publishing houses at a fraction of the cost. It's a more flexible medium, with no loss in quality' (2000). The first generation of the JIPAM e-journal appeared online in early 2000 and it was developed on the basis of static HTML files on a Unix platform. The web site contained basic content with little to no functionality.

1.1. THE NATURE OF THE RESEARCH PROBLEM

As part of their strategic planning for the continual development of JIPAM, the Managerial Board wished to take fuller advantage of more of the unique publishing opportunities that the Internet provides for e-journals such as global access combined with the functionality of different levels of access, speed of publication and reduced maintenance costs.

Several problem areas that were directly related to the static nature of the site's content were identified and are outlined below.

- Maintenance was a particular issue because of the growing number of hyperlinks required to maintain the static content. This leads to an associated problem concerning the effort required to ensure high levels of consistency were maintained throughout the numerous links.
- It was not possible for visitors to the JIPAM site to undertake online searches.
- A desire to be able to track the peer review process of new articles from the initial receipt of the draft article through to its publication.
- Although access to the full text of each article published in JIPAM was currently free, this may change in the future with the application of a fee for full access articles. Thus, the web application needed the functionality of a process whereby users were required to register (or subscribe) before gaining access to the full text articles.

1.2. RESEARCH OBJECTIVES

The overall objective of this research project was to redevelop the current first generation JIPAM e-journal web site into a dynamic web site addressing all of the above problem areas. In broad terms, the research project seeks to ...

- create a database framework in which JIPAM articles can be stored, searched on and retrieved, *and to also*
- develop an ASP front-end to provide dynamic access online to the JIPAM database.

1.3. RESEARCH METHODOLOGY

Unified Modeling Language (UML) formed the underlying methodology used to develop the JIPAM Web Site Dynamic Database System. The UML has evolved from the work of Booch, Jacobson and Rambaugh and it is a language for specifying, visualising, constructing and documenting software systems. It provides a 'unified industry-standard graphical language that enables ones to develop software models' (Vadaparty 2000).

Kobryn has said that UML 'is the software industry's dominant modeling language'. He goes on to say that 'UML is not only a de factor modeling language standard; it is fast becoming a de jure standard' (Kobryn 1999, p.29).

It also represents a collection of the best engineering practices that have proven successful in the modelling of large and complex systems. However, it is important to note that UML focuses on a standard modelling language, not a standard process. In fact, one of the advantages of UML is that it is independent of particular programming languages and development processess.

Although UML is evolving into a common and standardised methodology, some of the ideas and principles of each of the three originators (Booch, Jacobson and Rambaugh) remain unique.

Jacobson's ideas about UML, as published in his 1992 book: 'Object-Oriented Software Engineering' became the dominant methodological thinking used (with only minor amendments) in this project. Jacobson's view of UML is that system development is model building (with object modelling the central technique). 'The complexity needs to be handled in an organized way' and 'by introducing the complexity gradually in a specific order in successive models, we are able to manage the system complexity' (Jacobson 1992, p.113). Each of the models focus on a certain aspect of the system, see Figure 1 below.

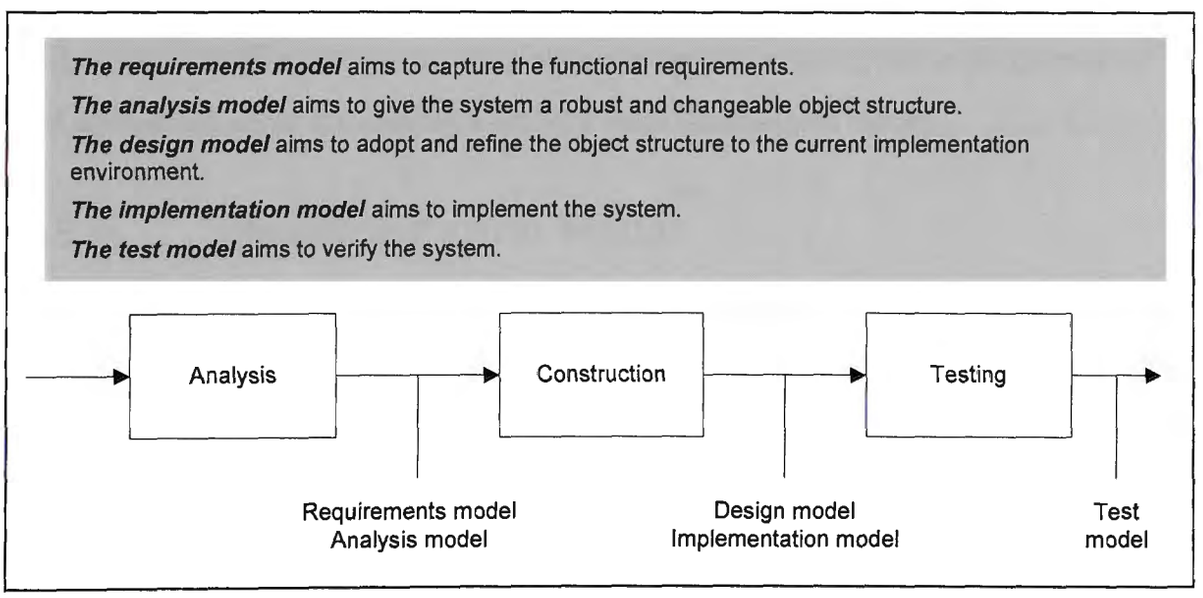


Figure 1. Jacobson's Five Basic Models (1992, p.113)

Hence, the requirements model is the starting point and it is derived from the system's requirement specification.

1.3.1. Requirements Model

In terms of the methodology followed in this research project, the requirements model was derived from discussions held with both John Roumeliotis (Managing Editor of JIPAM) and Stephen Young (member of the JIPAM management team and thesis supervisor). Use cases were then developed based on their stated requirements that formed the requirements model.

1.3.2. Analysis Model

The analysis model was effectively built directly from the use cases specified in the requirements model where the behaviour specified in the use case descriptions is distributed among three object types (interface, entity and control). The aim of the model is to form a logical and maintainable structure for the system.

1.3.3. Design Model

In Jacobson's methodology, the design phase is where the analysis model is translated into a model that will operate in the real world. Typically at this point, objects are determined. However, as ASP was the implementation language specified by the client for this project and as it does not support objects, the implementation was completed using the modules derived from the analysis model.

1.3.4. Implementation Model

The implementation model is the actual programming code which is the chosen solution to the problem as defined by the requirements model.

1.3.5. Test Model

Jacobson defines the test model as being 'developed to support the verification of the developed system' (1992, p.114). His methodology was followed through user tests to ensure that the system behaved in the same manner as defined in the use cases.

1.4. THESIS OUTLINE

This is a very project-based research thesis and extensive documentation of the modeling and development processes involved in the second generation of the JIPAM Web site has been included throughout the thesis.

Chapter one has introduced the research by presenting the nature and scope of the problem being investigated and the methodology followed.

Chapter two defines the Web environment and the constraints and challenges associated with developing a database-driven and dynamic site architecture.

Chapter three begins the redevelopment process using Jacobson's object-oriented methodology by outlining the requirements model.

Chapter four moves to the analysis model in Jacobson's development process. Two subsystems emerged from the modeling process — a viewing subsystem and a maintenance subsystem.

Chapter five discusses the design, construction and implementation issues associated with the development of the next generation of the JIPAM web site. Constraints in the application building process are discussed.

Chapter six draws the project to a close and compares the final status of the JIPAM web site with the objectives established in Chapter one.

Also included are the following five appendices.

Appendix A — CD-Rom containing the Second Generation JIPAM Web System

Appendix B — Programming code to set up the database

Appendix C — Programming code associated with implementing the web site

Appendix D — Programming code to implement the maintenance program

Appendix E — Administration Program Guide (50 pages).

2. WWW ENVIRONMENT

The World Wide Web (WWW) is an electronic environment where an Internet user runs an application program on their machine called a browser to obtain information from a information provider called a web server (host).

The WWW uses the hypertext transfer protocol (HTTP) which is an 'application-level protocol with the lightness and speed necessary for distributed, collaborative, hypermedia information systems' (RFC1945). It is based on a request/response paradigm. This paradigm implies a client/server relationship in that a client (request) retrieves information from a server (response). Two versions of the HTTP protocol are in use. HTTP/1.0 protocol has been used since 1990 while HTTP/1.1 is only a recent addition.

> *HTTP/1.0 does not sufficiently take into consideration the effects of hierarchical proxies, caching, the need for persistent connections, and virtual hosts. In addition, the proliferation of incompletely-implemented applications calling themselves "HTTP/1.0" has necessitated a protocol version change in order for two communicating applications to determine each other's true capabilities (RFC2068).*

In the main, it does not matter which of the two protocols are used for the JIPAM web system enhancement project because the IIS server interprets the messages and provides access to the information in the request headers. However, the version 1.1 protocol (RFC2068) is useful for determining information about users of the JIPAM web site.

Once the browser has the address of the site (or a document on the site), a request for the specified document is sent to the site. The layout of the message sent to the host is defined in the HTTP protocol. The host interprets the message and sends a response as an HTML stream that the browser processes. The way that the browser responds to the return message from the host depends on the content of the response message. The response is generally displayed in the client's window but there are conditions when this will not occur. Sometimes, the browser will attempt to save the response as a file or it may start another program that will interpret the response coming from the host.

The layout of the response message is also defined in the HTTP protocol as well as a number of supplementary RFC's that provide for additional features in the response message.

In the early days of the web, all document content was static. When a request for an individually identified document was received by a host, the document was located and returned back to the client as the response message. This approach was fine for content that did not change very much but it did not allow interactive responses nor did it provide functionality except for static linkages to other web pages. Even though it does have limitations, it is still used today for sites that are only changed very rarely.

However, when information provided by a host needs to be updated or when information is sent from the client to the host for processing, an alternate mechanism was required.

2.1. Common Gateway Interface (CGI)

The first attempt at processing dynamic content occurred when the common gateway interface (CGI) was defined. The CGI interface allowed programs to run on the server allowing the creation of a HTML page that would reflect the result of executing the requested program.

In the CGI interface, the host creates a number of environment variables containing information about the request. If additional information is sent from the client, it is written to the standard input device. The host then starts executing the requested program. The program examines the environment variables and reads the standard input device if appropriate and then writes its reply to the standard output device. The host collects the reply from the output device and returns it to the browser. Both these devices behave in the same way in both a UNIX and a WINDOWS operating environment which means the CGI interface was common to hosts running in either environments.

By using the CGI interface, the host server could interact with third party programs and as a result the nature of the Web changed dramatically. Now content, could be

dynamically created. One such example involves database programs responding to interactive queries of its content.

2.2. Internet Server Application Programming Interface (ISAPI)

When Microsoft released their web server program, they also introduced the Internet Server Application Programming Interface (ISAPI) as an alternative mechanism to run third party programs.

They did this because they considered that the CGI process was not very efficient in using resources on the web server running in the Windows operating environment. One example of the inefficiency generally cited is that of a web server that receives a large number of requests for exactly the same program to be run. To understand why this might be a problem requires some knowledge of the way programs run in the Windows Win32 environment. In this environment, there are two quite distinct approaches to starting another program.

- The first approach is to start the other program as a thread. A thread runs in the same address space as the starting process and has access to all its variable and environmental settings. The only additional memory the new thread requires is for its own program stack.
- The other approach is to create a complete instance of the other program. This approach uses the most system resources as each program instance requires it own memory allocation for all its needs including code space, variables, environmental settings and program stack space.

Starting a program as a thread requires far less memory than starting a program as a separate instance. Further, it also takes longer to start a separate process than it does to start a separate thread. One disadvantage of threads is that they must be part of the starting program. This requires the thread programs to be written as dynamic link libraries (DLLs). However, once the DLL is loaded in memory, it remains there until the host program unloads it. As a result, the next time it needs to run, very little time is required for the thread program to start.

Returning to the inefficient scenario cited earlier, it can be seen that it would be much more efficient in a Windows operating environment to run third party programs as threads and therefore gain resource savings on the host.

The Internet server application interface ISAPI was developed as a result of these issues.

Filters were another facility that came with the introduction of ISAPI on Win32 platforms. 'An ISAPI filter is a custom DLL that is in the same address space as the web server and is called by the web server in response to every HTTP request ... The filter then instructs the web server on how to handle the request. ISAPI thus allows you to customize your web server's response to specific types of user requests' (Keyton Weissinger 1999, p.5).

2.3. Active Server Pages (ASP)

Active server pages is a microsoft technology that leverages the installable filter aspect of their ISAPI interface. When a browser requests a file that has an ASP file extension, the web server passes the requested document to the active server page filter ASP.DLL. The ASP.DLL routines process the passed file to produce HTML output that the web server then passes back to the browser.

The ASP file can contain both program code and HTML formatting code. The program code is generally a form of the visual basic code popular on Microsoft platforms. It is interesting to note that the code does not have to be visual basic. It could also be Perl, for example, or any other scripted (interpreted) language. The language chosen depends on how the ASP filter is setup, that is, what scripting language interpreter DLL the ASP DLL should load to process the incoming request. The default setup of the visual basic scripting language will be the only one considered for the JIPAM web system enhancement.

ASP is also not an object oriented language. While it supports the use of objects, these objects are Active X objects installed into the operating system. There is no language support for directly specifying objects using ASP.

2.3.1. The Visual Basic ASP Environment

An ASP page is a visual basic program that can also contain HTML formatting code. The visual basic code in an ASP page is separated from the HTML code by beginning and ending markers, <% and %> respectively. The ASP script interpreter reads the file and executes the visual basic code. When it comes to HTML formatting code, it writes the HTML directly to the output message. It then continues on with the visual basic code program sequence until it reaches either the end of the page or a message terminating the output page.

When a host site is configured to use the visual basic scripting language, there are a number of components/objects (see list below) created automatically in that environment which are accessible by direct reference in ASP pages.

- Application Object
- ObjectContext Object
- Request Object
- Response Object
- Server Object
- Session Object
- Application Object

An IIS web site has a hierarchial directory structure which consists of a base directory called the WWW root directory. A chain of children directories are under this directory and when they are combined they form the web site.

An application in terms of the IIS web server consists of all the documents/files that are accessible to an Internet user from a particular directory and includes all the contents of that directory and its subdirectories. The particular directory is marked by the presence of a file named **GLOBAL.ASA**.

The Application Object is initialised by the IIS as a result of the presence of the **GLOBAL.ASA** file the moment the first Internet user accesses a document from directory containing the **GLOBAL.ASA** file or any of its children. The IIS also runs the application ON_Start event code if that code is present in the file. This event code

can itself initialise variables and other objects that should be visible to all users of the application and persist for the duration of the *Application Object*. The **GLOBAL.ASA** file can also contain ON_End event code which is run when the *Application Object* is stopped. The *Application Object* can be stopped from the Microsoft Management Console but is more generally stopped when IIS is shut down.

The ObjectContext Object was made available in version 2.0 of Active Server Pages. It enables the concept of a database transaction to be implemented by allowing a transaction script to be created. Thus, it is possible to create one code section of ASP code that removes a record from a database table while another adds a record to another table. If both sections of code complete successfully, the database will be updated permanently with both changes. If either or both fail, the transaction will be rolled back, that is the database would be returned to the state which it was in before either of the two code sections commenced. This transaction support in IIS is based on Microsoft Transaction Server technology which is built into the IIS.

The Request Object is created for every request message that an Internet user sends to the ASP application. It allows program access to the HTTP request header and body without the ASP application having to parse the incoming message.

The Response Object is created to represent the HTML message that will be returned to the Internet user. For every *Request Object* created, there is a corresponding *Response Object*. Like the *Request Object*, it allows programs access to the HTTP header and body of the return message.

The Server Object represents IIS itself and exists for the life of the web server. It provides miscellaneous functions available to all applications and the *CreateObject* method is the most important. It allows ASP applications to use third party components as though they were part of the ASP environment themselves. A number of these components are provided when the IIS program is installed. The ActiveX Data Object (ADO) is the most important of the components for the JIPAM Web System Enhancement because it allows access to databases.

The Session Object is only created if the ASP application makes reference to the object directly. Its role is to hold data variables that are applicable to an individual

user. The *Session Object* for an individual user is identified by its SessionID value. This value is a 32 bit number that the web server generates when the object is created. The SessionID value is sent as a temporary cookie to the client's browser. The cookie exists only until the client closes their browser software.

2.3.2. File Uploads

When creating a dynamic web program, it would be of benefit if clients could send data from their machines to the web server as this feature would enable clients to update databases dynamically. The functionality to send data from the client machine is built into the HTML language with its <FORM> and </FORM> tags. The original specification for the input element tag included a variety of field formats but it did not include files.

Since then, the HTTP definition and the HTML specification has been extended to allows files to be sent to the web server as part of the request message. This specification is described in RFC1867.

This feature has not been provided as part of the ASP DLL. The extraction of file data from the request message has been left to other third party suppliers to implement.

2.4. Web State

As discussed previously, HTTP is based on a request/response paradigm which unfortunately has a negative side effect.

After a web server receives a message, it formulates and sends back a response. It then promptly forgets that it is has just processed a message. When the server next receives a request (which could, quite possibly. come from the sender of the previous request) and it subsequently responds, it also forgets about that message as well. This sequence of the web server receiving, responding and forgetting is continuous which means that there is no history of previous messages maintained. Thus, it is said that the web server does not maintain state or is stateless.

Being stateless can cause problems. In the scenerio where a web site requires customers to pay for access rights, there needs to be some mechanism to identify users who are registered customers. This is a simple example because users could be required to identify themselves as registered customers when they first connect to the site. However, as the web server is stateless and does not remember its message history, it will not recognise the next message as being from the same registered customer. Instead, the incoming message is seen by the server as a new message and the customer must again identify themselves to gain access to the site. The need to continuously identify as registered customers would quickly irititate customers.

However, there is a mechanism in the HTTP 1.0 protocol that provides for persistent connections between the web server and the client software (RFC1945). Its use is limited though.

> *The Hypertext Transfer Protocol 1.0 allows client browsers to send Keep-Alive messages to proxy servers. These messages basically tell the proxy server to maintain an open connection with the requesting client. However, these connection requests are often unrecognised by the proxy server. This problem in the proxy server results in a hung connection between the proxy server and the requested web server. In a nutshell, maintaining connections with web servers is prone to error and thus unreliable in HTTP 1.0, still by far the protocol most commonly used by client browsers (Keyton Weissinger 1999, p.122).*

A mechanism was needed to provide state information between the sending one page and the next to a client.

2.4.1. Cookies

Netscape Communications Corp developed the term 'cookies' to refer to their Persistent Client State Mechanism that they saw as a way of overcoming the lack of state in the HTTP protocol.

The concept behind cookies is quite simple. They are small pieces of information that the web server could store on the client's machine and are name/value pairs created by the server and included in the body of the response message. A client's browser would then include these name/value pairs in the body of any request message that was then sent to the server originating the message. In this way, the

server could maintain appropriate historical information associated with requests from one client.

However, as well as the name/value pairs associated with a cookie, there are also additional properties included that determine the way the client's machine should process them.

- ***Expires*** which tells the browser when it can delete a cookie from the client's machine. If no date is specified, the cookie lasts for the duration of the current HTTP session.
- ***Domain*** which tells the client when it should provide the cookie back to the web server. The domain of the web server must agree with the domain setting in the cookie. This ensures that a cookie provided by one web server does not get included in messages to another web server.
- ***Path*** which tells the client which application on the web server the cookie belongs to. This also ensures that unrelated applications running on the same web server do not receive cookies from another application.
- ***Secure*** which tells the client browser to provide the cookie if the browser and the web server are using the secure HTTP protocol.

There are two associated issues that have a bearing on the use of cookies. One is that at the current time, all browsers do not support cookies. The second issue relates to the ability for web users to disable cookies in their web browsers. If cookies are disabled in this way, the issue of state arises again.

2.5. Web Security

> *Security is an issue that most people would just as soon ignore – whether it is on the Web, in your car, at home, or anywhere else. The essence of security is keeping things safe: in the case of the Web, that means information. Safety is defined as many things: keeping the wrong people out, letting the right people in, allowing only certain people to enter or modify certain data elements, determining conditions under which security procedures can be modified, and on and on and on. In practical terms, security often seems to be a barrier or roadblock erected between the user who wants information and the information provider who wants people to have information (Feiler 1999)*

The web server software IIS runs in an NT environment so it is necessary to understand the relation between the web server security system and the NT security system.

In the NT operating system, everything that a user accesses and uses is controlled by the NT security system, for example, a user must be first logged on to gain access to the desktop in a NT environment. The user needs to enter a user account and a password that the security system checks before access is allowed. The level of access is determined by the permissions the NT administrator allocated directly to that user or the groups to which that user has been assigned.

This degree of security is fine if information or services provided on the machine is restricted to a group of individual users. In the web environment, information is available to the global community and allocating individual user names and passwords is not feasible.

An alternative could be to use a guest account but as Homer indicates 'guest (or anonymous) accounts are generally not a good idea within Windows NT itself, and are usually shunned by network administrators because they can provide opportunities for hackers to log onto the local machine, and then attack the network' (1998, p.188).

IIS overcomes this issue by creating a user account (IUSR_*machinename*) with a randomly generated password that it uses when it is installed. When a Internet user requests information from the web server, the web server retrieves the information from the NT environment using the IUSR_*machinename* account. Thus, the access that IIS is granted through this account can be controlled in the same way that other users of NT are controlled. In the initial setting, this user is made a member of the Guest group and therefore has access to the same resources that the Guest group has.

2.5.1. Authenticating Users

There are situations where anonymous access using the IUSR_*machinename* account does not provide access to all resources on the NT machine. Alternatively, access to certain parts of the web site may need to be restricted to certain users (for example, those that pay a fee).

There are two ways to restrict or enhance access on the web site. The first, which has already been discussed, concerns the allocation of an individual user account. The second way is to have user programs running under IIS. These programs control access by displaying web pages which accept usernames and passwords.

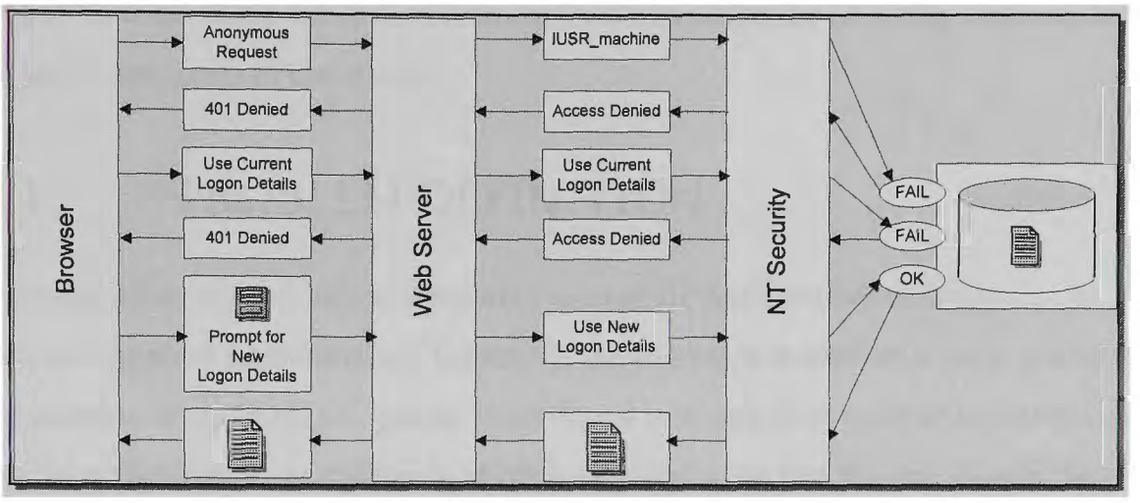


Figure 2. Authentication with BuiltIn Browser/Server Methods (Homer 1999, p.202)

Homer discusses the first option in great detail and his diagram reproduced above summaries the logical flow of messages that pass between the web server and the users's browser when trying to access resources not available to the IUSR_*machinename* account. Most of the time these messages are transparent to the user, especially when using Microsoft's browser (Version 3 onwards) which has support for the NT security system built in.

The second method is the simplest to implement and it is probably the method of choice when the level of security provided by the NT security system is not needed or the NT security system does not provide the degree of control required. The second scenerio (where the NT security system does not provide the degree of control required) arises when database access is involved. The NT security system will determine whether a user has rights to access the resource (database) but after access is given, it is not concerned with the user's actions regarding the data within the database. In a dynamic web environment, this level of security needs to be provided by the underlying database software.

3. REQUIREMENTS MODEL

The requirements model is built from the perspective of the user and it comprises use cases and their associated actors. A component of the object-oriented approach is the need to be clear about the system boundary with an emphasis on being explicit about the scope and goals of the system.

3.1. PROBLEM DEFINITION

Victoria University publish an electronic journal JIPAM (Journal of Inequities in Pure and Applied Mathematics). Currently, the journal is hosted on a Unix platform and consists of static HTML pages. Each time a new article or volume is placed on the site, a significant maintainence effort is required to update the site. In addition, the site does not provide search capabilities on the article contents because of its static nature.

The editor of the journal wants the site to be easier to maintain. This is to be accomplished by migrating the site to a database solution thereby allowing search capabilites to be provided. He also wants to restrict access to the full text of articles published in JIPAM by requiring users to login before they are able to retrieve the full text. He has a long-term objective of charging users a fee for such access (e.g. access to the full text of the journal articles online).

A further requirement is to use only minimial Javascript as a number of Web users still have relatively primative browsers. This last requirement also applies to the level of HTML to be used.

The editor also required Microsoft SQL 7 to be used as the target database with Microsoft Internet Information server providing the web functionality.

3.2. SCOPE OF THE PROJECT

It is critical to be clear about the boundaries of the system. Table 1 defines the scope of the JIPAM Web Site Dynamic Database System (Phase 1).

YES	NO	ITEM
✓		Storing user (subscriber) details.
✓		Searching the Journal using a range of criteria.
✓		Accessing the full text of published articles
✓		Storing all published articles
✓		Storing all unpublished articles while undergoing peer review
✓		Maintaining the system
✓		Storing Editor details
	×	Tracking the peer review process for unpublished articles.
	×	Charging a subscription fee to access full text articles

Table1. Scope of the JIPAM Web Site Dynamic Database System

In determining these boundaries, all of the user's requirements have been considered to be within scope except for the last item, that is, the tracking of the peer review process of unpublished articles. This functionality has been considered beyond the range of the current project and it remains to be developed and implemented as a future enhancement of the system.

3.3. STAKEHOLDERS

There are six groups of people interested in the system. These groups are defined as being stakeholders in the system and they are described as follows.

1. ***General Users*** are people from the global community who wish to either browse or search the JIPAM web site as a source of information relating to current mathematics research. As well as browsing through the table of contents for each issue of the journal, they may also be interested in searching the journal using a range of defined criteria (such as author name and/or keyword). The search results provide a summary of specific articles complete with an abstract. However, to access the full text, general users must be registered as subscribers.
2. ***Subscribers*** are general users who want access to the full text of articles.
3. ***Authors*** are professionals within the mathematics discipline who have undertaken research and who now seek to publish their research findings.

4. ***Editors*** are also professionals in the field of mathematics who are generally recognised by others in the international maths community as being leaders in the field.

5. ***The JIPAM Web System Administrator*** is a member of the Department of Communication and Informatics, Victoria University and has administrator responsibilities as part of their functional duties at the university.

6. ***The Managerial Board*** seeks to disseminate research results quickly to the international maths community using an efficient and up-to-date publication process which maximises the delivery benefits of Internet technology.

The term 'stakeholders' refers to actual roles that individuals and/or groups play in their interactions with the JIPAM web system. A person can be a member of more than one of these stakeholders' groups, for instance, the author of a specific journal article would most likely also be a subscriber and they may also undertake the role of journal editor. Five of these six groups are direct users of the system with the Managerial Board being primarily involved with strategic concerns rather than operational issues. Figure 3 below provides an overview of the JIPAM Web Site stakeholders and their goals.

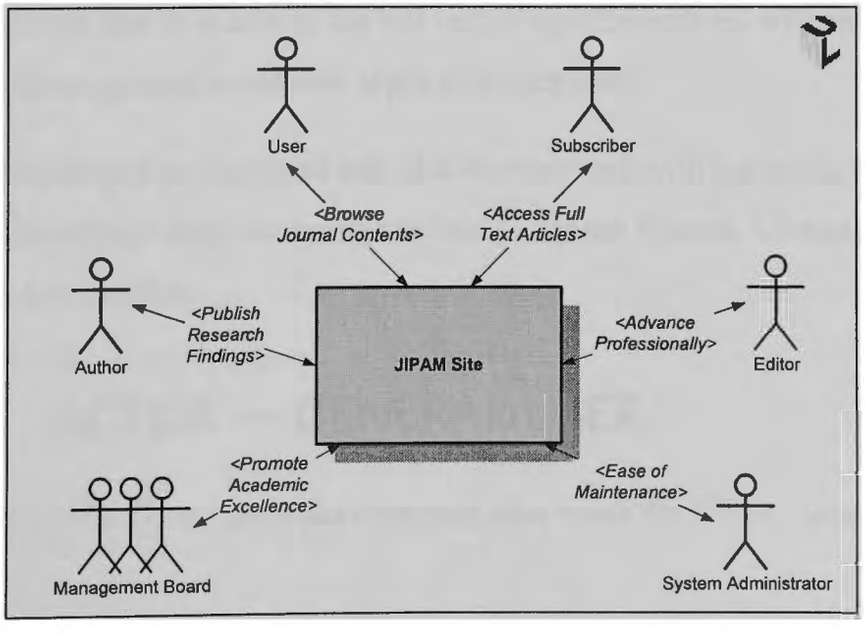


Figure 3. JIPAM Web System — Stakeholders and Their Goals

3.4. USE CASES

'Use cases provide a way of describing the external view of the system and its interactions with the outside world' (Fowler nd, http://www2.awl.com/cseng/titles/0-201-89542-0/techniques/useCase.htm). According to Jacobson a use case is ' a behaviourly related sequence of transactions' performed by a user 'in a dialogue with the system' (1992, p.127).

Each interaction with the system performed by a user will result in a use case scenario. The combination of all use case scenarios defines the behaviour of the system from the user's point of view.

3.4.1. General

In considering use cases for this system, five actors (as described in 3.4) have been defined.

- General User
- Subscriber
- Author
- Editor
- System Administator

The URL for the JIPAM web site will be generally available to all five actors with some restrictions, that is access to the full text of specific articles will only be available to those general users who login as subscribers.

Access to administration functions will also be restricted with the menu selection for administration actions only becoming available after the System Administrator has logged in as a subscriber.

3.5. ACTOR — GENERAL USER

The group 'General Users' includes everyone who visits the JIPAM Journal web site.

3.5.1. USE CASE 1— Collect Statistics

Very basic statistics will be collected from everyone who visits the site. The statistics will be extracted from the request header provided by the general user's browser.

Typical data items could include the visitor's language, their country of origin and the type of browser the visitor is using.

3.5.2. USE CASE 2 — Search JIPAM

General users need to be able to search the JIPAM Journal site using the following five search criteria.

- Search on the author's name
- Word/s in the article title
- Keyword/s
- Classification code/s
- Any word/s in the article abstract itself

To search, general users will be presented with an entry screen where they enter the search criteria (either the whole text, or part thereof). There will be a submit button on the screen which, when selected, will initiate a search of the database. The results of the search (based on the criteria entered) will then be displayed on the screen.

The search results will appear in the form of a list of the titles of all the articles held in the JIPAM database that match the search criteria specified. In addition to the article titles, the relevant author's name/s and the volume/issue details are also listed. There will be a limit on the number of entries that the search returns on a single screen to prevent the whole database being displayed.

Each of the three items in the search results screen will be hyperlinks to more detailed information and they are discussed separately below.

3.5.2.1. Article Title

The article title will be a hyperlink to more detail about the article (the article details which provides the following information).

- Article title

- Article author details (the name and contact details which contain a hyperlink to their email address and also their individual web page, if one is available)
- Author's institution (their affiliation)
- Date article draft received
- Date article draft accepted for publication
- Accepting editor's name
- Abstract
- Keyword/s
- Classification code/s

If the full text is currently available (that is, it has either been published or is currently awaiting inclusion in the next volume/issue to be published), hyperlinks to it will be provided. The full text will be available as PDF files in two formats: one for viewing on the screen and one for printing.

However, if the user is not logged in, the hyperlinks will not be displayed. Instead, instructions on how to subscribe or login will be provided (see Section 3.5.3).

3.5.2.2. Author Name

The author's name will be a hyperlink to their publication list. The details of the author (the name and contact details which contain a hyperlink to their email address and also their individual web page, if one is available) will be provided.

A list of all articles (showing the article title and volume/issue details) associated with this author will be displayed. The article title will be a hyperlink to the article details (see above) and the volume/issue details will be hyperlinked to the table of contents for that specific volume/issue.

3.5.2.3. Volume/Issue

The volume/issue will be a hyperlink to the table of contents for the selected volume/issue of JIPAM. The publication number details (volume number and issue number) will be shown followed by a list of all articles in the publication. The article list will show the article title (a hyperlink to the article details) and the article author (a hyperlink to all the articles written by that author, e.g. their publication list).

A list of all of the JIPAM volume/issues currently published will also be provided at the end of the list of articles. Each publication number will be hyperlinked to its table of contents.

3.5.3. USE CASE 3 — Subscribe to JIPAM

To access the full text of articles published in JIPAM, general users must become subscribers. This will be achieved through the login function which displays a login screen and information about the subscription process.

If a general user has not previously regististed as a subscriber, there will be a hyperlink within the information displayed on the login screen. This hyperlink displays an input screen with a submit button which the general user will be required to complete. Some of the data will be compulsory (shown with a asterisk) while other data will be optional. The compulsory data will be the user's email address (which is also their login name) and a password.

The optional details include:

- Name
- Physical address
- Institution (their affiliation)
- Individual web page address

There will be a submit button on this screen which the general user will select after they have entered all the required details.

The details entered will then be checked to confirm that all compulsory fields have been completed. The email address will be unpacked to see if it is a valid email address. The details will be entered into the database and the user will be logged in.

3.5.4. USE CASE 4 — Display Current Volume/Issue

General users will be able to access details of the current volume/issue. A list of the contents of the current volume/issue showing both the article title and author/s will be displayed. Each article title will contain a hyperlink to the article details (see 3.5.2.1.) while each author name (see 3.5.2.2.) will also contain a hyperlink to all the articles written by that author (their publication list).

A list of all other volume/issues currently published will also be provided at the end of the list of articles. Each publication number will be hyperlinked to its table of contents screen.

3.5.5. USE CASE 5 — Display Articles yet to be Published

A list of all articles (article title and author/s) that have been accepted but not yet published in JIPAM will be available to general users. Each article title will contain a hyperlink to the article details (see 3.5.2.1.) while each author name (see 3.5.2.2.) will also contain a hyperlink to all the articles written by that author (their publication list).

3.5.6. USE CASE 6 — Display List of Editors

A list of all of the editors of JIPAM in alphabetical order will be available to general users. The editor's name and their institution details will be provided followed by either one or two symbols; the first symbol (an envelope) will be a hyperlink to the editor's email address while if the other symbol (a globe) is shown, it will be a hyperlink to the editor's individual web page. The second symbol will only be displayed if the editor has their own individual web page.

3.5.7. USE CASE 7 — Provide Feedback

By using a text entry form, general users will have the facility to provide feedback to the JIPAM Administrator. The data entered by the user will be stored in the database. An email facility to contact the JIPAM Administrator will also be provided.

3.5.8. USE CASE 8 — Static Pages

There are several activities (as listed below) that a general user can perform that will provide access to information about JIPAM which are based on static HTML pages.

- *Aims and Scope* will describe the aims and scope of JIPAM.
- *Contact* will indicate how to contact the JIPAM editorial team.
- *Copyright* will discuss copyright issues for JIPAM.
- *Management* will list the members of the JIPAM management board.

- *PDF Files* will detail the format of full text articles provided for publication in JIPAM.
- *What's New* will provide the latest information about JIPAM.

3.6. ACTOR — SUBSCRIBER

Subscribers are general users who have additional access rights which allows them to download the full text of JIPAM articles. Note that general users will need to be logged in as a subscriber before they can access the full text.

3.6.1. USE CASE 9 — Login as a JIPAM Subscriber

General users need to login as subscribers before they can access the full text of articles. A login screen is displayed with information about subscribing (see 3.5.3). There will also also be an input area for them to enter their login details plus an associated login button to complete the login process. To login, subscribers will need to enter their username (which is their email address) and their password.

As part of the validation process, the subscriber's details will be checked to ensure that their subscription to JIPAM has not lapsed (that is, the annual date is still current). Given that in Phase 1 of the system development, there will not be any subscription charge, the renewal date will be updated automatically.

The login process will either be successful or not — both outcomes are now discussed in more detail.

3.6.1.1. Login is Successful

If the login details entered are valid, the subscriber will receive a confirmation message that they are logged in. At the same time, there will be a hyperlink provided to allow subscribers to update their details, if they feel it necessary to do so.

Once logged in, subscribers then have access to the full text of articles that have been published in JIPAM or that are awaiting publication.

3.6.1.2. Login is Unsuccessful

If the login details are not recognised as being valid, the subscriber will be advised that they cannot be logged in. Another input form will then be provided for them to

try again. For times where they have forgotten their password, this form will also contain a hyperlink where they can request their password to be emailed to them. Another hyperlink will also be provided to enable them to become a subscriber.

A count of the number of times that a general user attempts to login as a subscriber will be maintained for each session. A successful login will reset the default count for the current session. The system will only allow three attempts to login. If the general user is still unsuccessful after their third try, no further attempts to login (and thus gain access to the full text) will be allowed during the current session.

3.6.2. USE CASE 10 — Change Subscriber Details

This option will only be available if the subscriber has already successfully logged in. Their current subscription information will be displayed and they will then be able to change the following details.

- Email address (also their username)
- Password
- Name
- Physical address
- Institution (their affiliation)
- Individual web page address

Once the changes have been entered, the subscriber selects the submit button which will update the JIPAM database. A message will then be displayed that confirms the changes were successfully carried out.

3.7. ACTOR — AUTHOR

Authors submit draft articles to JIPAM for publication which then undergo an editorial review to determine acceptability for publication in JIPAM.

Tracking this editorial review process will be outside the scope of Phase 1 of this project.

3.8. ACTOR — EDITOR

Are respondible for reviewing draft articles and applying the relevant publications standards to the contents of all volumes/issues of JIPAM. Tracking this editorial review process will be outside the scope of Phase 1 of this project.

3.9. ACTOR — ADMINISTRATOR

Provision needs to be made for administration and maintenance of the JIPAM system. Thus, there is an actor known as the JIPAM Administrator.

3.9.1. USE CASE 11 — Login as Administrator

The JIPAM Administrator must login as a subscriber before any system administration and/or maintenance can be done. If the correct login details are entered and subscriber has administrator rights, an extra menu option will be available to allow database maintenance activities.

3.9.2. USE CASE 12 —Change Default Values

An entry form will be displayed that shows the existing default values and these values can be changed by simply entering new values into the form and clicking the accept button.

The actual system defaults are now discussed in greater detail.

3.9.2.1. Grace Period

Although Phase 1 of the system has been developed based on the premise that subscription will be free, some functionability towards moving to a charge basis needs to be provided within Phase 1. Hence, subscriber registrations will provide full access rights for a defined period of time (12 calendar months from the initial subscription date).

The grace period is the amount of time after the annual renewal date that the customer has available to them to renew their subscription. If they do not do so, their subscription details will be deleted from the database. In Phase 1 of the project, the renewal date will be automatically updated.

3.9.2.2. JIPAM Administrator Email Address

To facilitate communication with JIPAM administrator, there is facility within Phase 1 for the system to authomatically generate an email message addressed to the administrator. Thus, the default email address for the administrator must be known to the system.

3.9.2.3. Default Login Count

As a security measure, there will be a limit to the number of times (in one session) that a general user can attempt to login as a subscriber (to gain access to the full text). The default value within the Phase 1 is set to a maximum of three attempts.

If the count is exceeded, the user will be then locked out for the duration of their current session.

3.9.2.4. Maximum Search Result Count

There will be a limit on the number of records that will be displayed as a result of a search. Within Phase 1, the default value is 100 records.

3.9.3. USE CASE 13 — Create a New Article

New articles for JIPAM will be created in a number of stages.

3.9.3.1. Basic Information About the Article

Firstly, a data input screen will be displayed for the administrator to enter basic data about the article (see the following list).

- Article number
- Article title
- Date article draft received
- Date article draft accepted for publication
- Accepting editor's name
- Abstract
- Available flag (provided to ensure that an article does not appear in article lists until all details about the article are acceptable to the administrator.

When these fields have been filled out, an accept button will add these details about the new article into the database. The system will validate the article number and the two dates.

The administrator will then be presented with a series of screens that will accept the following data elements for a new article.

- Authors
- Keywords
- Math codes
- Full text PDF files

3.9.3.2. Authors

A screen will be presented to the administrator showing a list of all the subscribers who are also authors within the JIPAM database . The administrator will then select the author/s for the new article in the required order. A select button will update the article on the database with these authors.

3.9.3.3. Keywords

A screen will be presented to allow the administrator to enter keywords for the new article into the database. When the accept button is selected, the keywords will be written to the database.

3.9.3.4. Math Codes

A screen will be presented to allow the administrator to enter the math codes for the new article into the database. When the accept button is selected, the math codes will be written to the database.

3.9.3.5. Full Text PDF Files

A screen will be presented to allow the administrator to select the full text PDF files for each of the two formats. When a file is selected, its contents will be written to the database.

3.9.4. USE CASE 14 — Change the Article Details

There are a number of reasons why it may be necessary for the administrator to have the facility to change the details of an article in the database. For instance, there may have been a mistake in one of the data entries when it was created. Also, an article may have originally been flagged as unavailable. In such a case, the unavailable flag will need to be changed.

The administrator will have access to a screen that displays a list of all of the articles in the database. The administrator then selects the article that needs modification.

An entry form will display all of the basic details relevant to the selected article that can be changed (see list below).

- Article no.
- Article title
- Date article draft received
- Date article draft accepted for publication
- Accepting editor
- Abstract
- Available flag

The existing entries are shown for each field and they can be changed by simply typing in the new data and then clicking the submit button.

The administrator will then be presented with a series of screens to allow changes to be made to other data elements associated with the article.

3.9.4.1. Authors

A screen will be presented to the administrator that will display the existing authors on an article as well as a list of all other authors with the JIPAM system. The administrator will be able to remove an existing author, select an author from the main list and reorder the authors in the selected list. When completed, the accept button will write the changes to the database.

3.9.4.2. Keywords

A screen will be presented to the administrator that will display the existing keywords. The administrator will be able to add new keywords or delete existing keywords from this list. When completed, the accept button will write the changes to the database.

3.9.4.3. Math Codes

A screen will be presented to the administrator that will display the existing math codes. The administrator will be able to add new math codes or delete existing math codes from this list. When completed, the accept button will write the changes to the database.

3.9.4.4. Full Text PDF Files

A screen will be presented to the administrator which will show the existing PDF files assigned to the article. The administrator will be able to delete existing formats. They will also be able to select a PDF file for any of the two formats not already in the database. When a file is selected, the contents of the file will be written to the database.

3.9.5. USE CASE 15 — Add a Subscriber

There may also be times where it is necessary for the administrator to manually add a subscriber to the database, for instance, the web subscription process may be unavailable for whatever reason.

The administrator will enter the subscriber's details into an input form (see 3.5.3. for more details about the data required). When submitted, various details will be validated and then the details will be written to the database.

3.9.6. USE CASE 16 — Change a Subscriber's Details

Subscriber details may also need to be changed by the administrator at various times. In particular, the administrator needs to be able to change the role of subscribers, for instance, a subscriber may become an author or an editor.

A screen listing all the JIPAM subscribers will be displayed to the administrator who then selects the subscriber whose details need to be changed. An input screen is then displayed which provides the subscriber's existing details. However, in addition to the fields that a general user completes as part of the subscription process, the following additional fields are also displayed.

- Institution
- Role
- Renewal date
- Start date

The administrator makes the necessary changes and the updated data is then written to the database.

3.9.7. USE CASE 17 — Delete a Subscriber

It is possible that the administrator will need to delete a subscriber before the automatic process associated with the end of the subscription period.

A list of the subscribers is displayed to the administrator who then selects the subscriber they wish to delete. The subscriber's details are displayed for the administrator to confirm the deletion.

3.9.8. USE CASE 18 — Update a Subscriber's Subscription Date

The renewal dates for all subscribers will be checked to identify those that have expired. In Phase 1, there is no fee being charged for subscription so all renewal dates will be automatically updated if they have used the system in the last three months. If they have not used the system in the preceeding six months, they will be removed from the system.

3.9.9. USE CASE 19 — Create a New Volume/Issue

The administrator will need to be able to create new volume/issues on an ongoing basis. An input screen containing the following text boxes will be available to the administrator.

- Volume no.
- Issue no.
- Issue year
- Issue date

A submit button will be provided and the date entered will be validated and a new entry will be made into the database.

3.9.10. USE CASE 20 —Add Articles to a New Volume/Issue

Once articles are accepted for publication, they are flagged as new papers. When appropriate, these new papers need to be added to the contents of the next volume/issue to be published (the new volume/issue).

The administrator will have access to a screen which displays a list of the articles associated with the new volume. In addition, the list of articles that have not yet been associated with an issued volume are also displayed. The administrator will be able to select those articles that will be included in the new volume/issue. This process will change the status of these specific articles from being unpublished to published.

The administrator will be able to also manage the order that the articles appear in the table of contents of the new volume/issue through this screen.

When the selection and order of the articles is completed, the database will be updated with the table of contents of the new volume/issue.

3.9.11. USE CASE 21 — Retrieve Usage Statistics

At some point, the administrator will want to examine the statistics the system gathers automatically from users to the site. A screen will be presented to the administrator that lists the statistics available to them. The administrator will select the desired statistic and a further screen will be presented to them showing the statistic data gathered.

3.9.12. USE CASE 22 — Add a New Institution

Because a list of institutions is maintained within the database, there needs to be facility for the administrator to add new institutions to the list. Thus, there is an input screen available to the administrator to complete with the following details.

- Name of the institution
- Full address
- Telephone no.
- Fascimile no.

There will be a submit button on the screen and when it is selected, the entered details will be validated and a new entry made to the database.

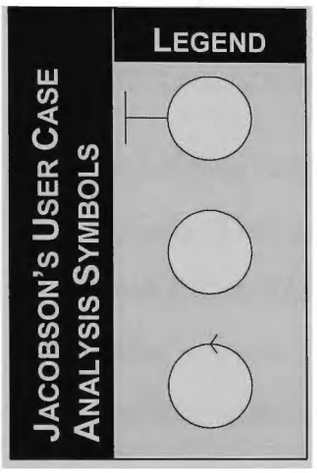
3.9.13. USE CASE 23 — Modify an Institution's Details

It is possible that the details of a specific institution may change, e.g their telephone or fascimile number. The administrator has access to a screen which lists all the institutions in the database. They then select the institution whose details they wish to change.

A second screen will appear that will show the details in the database for the selected institution. The administrator can modify the selected fields and when completed the updated details will be written to the database.

4. ANALYSIS MODEL

The next stage of the Jacobson visual modeling process is the analysis model which aims to form a logical and maintainable structure for the system. It is derived from an analysis of each use case followed by a distribution of the behavior of each use case into three object types (interface, control, entity) and the interactions between them.



- ***Interface objects*** are responsible for functionality that is directly dependent on the system environment. As described by Jacobson et al, 'The task of an interface object is to translate those events in the system that the actor is interested into something that can be presented to the actor' (1992, p.176)
- ***Entity objects*** are used to model the information that the system will maintain over a period of time (Jacobson et al, 1992, p.184). Generally, this period of time will be longer than the length of time that is covered by the use cases. This means that these objects represent the persistent data of the system.
- ***Control objects*** are typically used to model all behaviour that cannot be placed into either of the other two objects.

4.1. USE CASE ANALYSIS

Although there are 22 use cases, several have been combined during the analysis stage in instances where they have some common elements. Thus, the following 15 diagrams were created from analysis of the use cases.

4.1.1. Collect/Retrieve Usage Statistics

Figure 4 concerns the statistics gathered in relation to users of the JIPAM web site and incorporates Use Case No. 1 and 21.

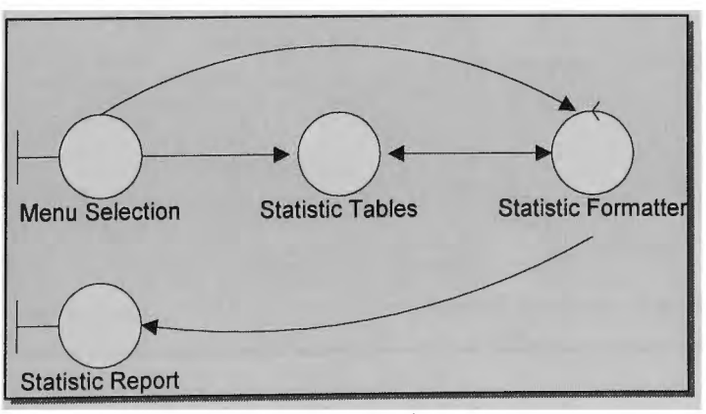


Figure 4. Collect/Retrieve Usage Statistics

The remaining 14 diagrams have been created on similar principles with the next two diagrams representing the search functionality and the subscribe/login functionality which are the most complex activities within the system.

4.1.2. Search JIPAM

Figure 5 represents the search functionality of the JIPAM web system as outlines in Use Case No.2. The entity objects within this diagram ies (i.e. Default Values and Database Tables) are shown more than once in order to reduce the number of interconnections and therefore keep the diagram simple.

According to the use case specification, each of the three interface objects (i.e. Articles Details, Author Publication List and Volume Contents List) would know of each other. The relationships that each of these interfaces would have with each other's control object does not appear in the diagram in order to ensure the diagram itself remains clear and simple.

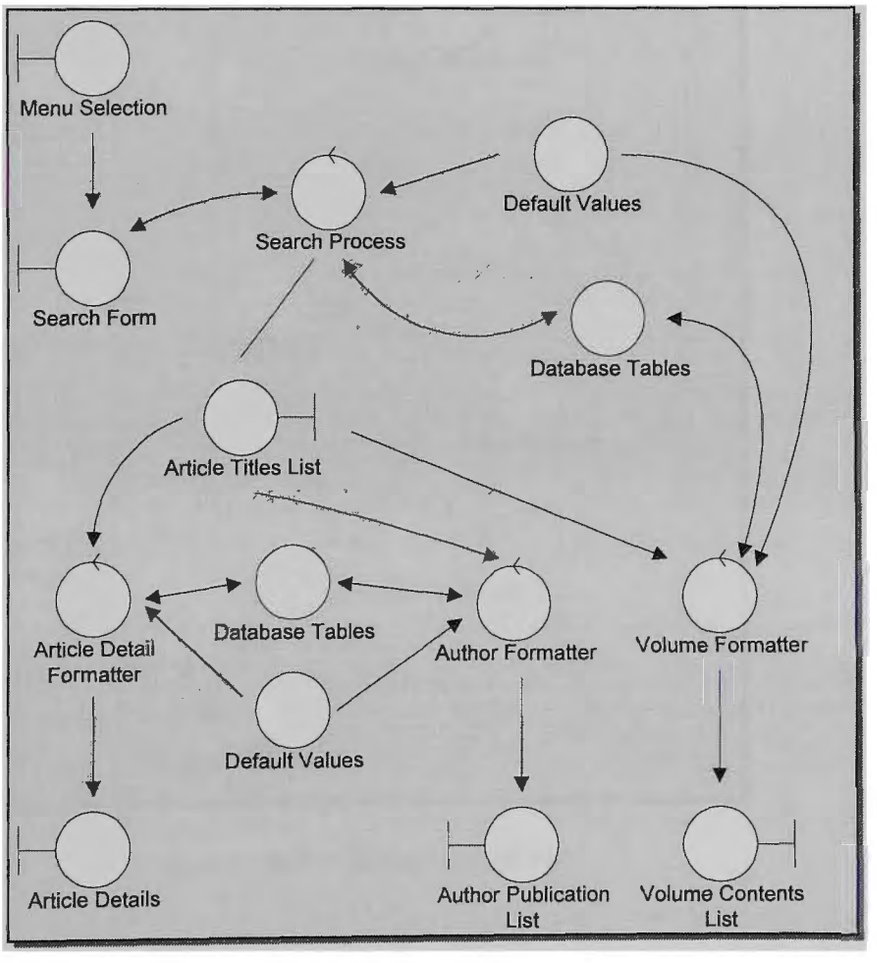


Figure 5. Search JIPAM

4.1.3. Subscribe to JIPAM/Login as a JIPAM Subscriber/Change Subscriber Details

Similiarly, the requirement for users to register as subscribers and then login as a subscriber before gaining access to the full text articles has meant that the analysis model shown as Figure 6 below is quite complex. The following use cases have been incorporated into one analysis diagram to condense the number of separate analysis diagrams.

- *Use Case 3.* Subscribe to JIPAM
- *Use Case 9.* Login as a JIPAM subscriber
- *Use Case 10.* Change subscriber details

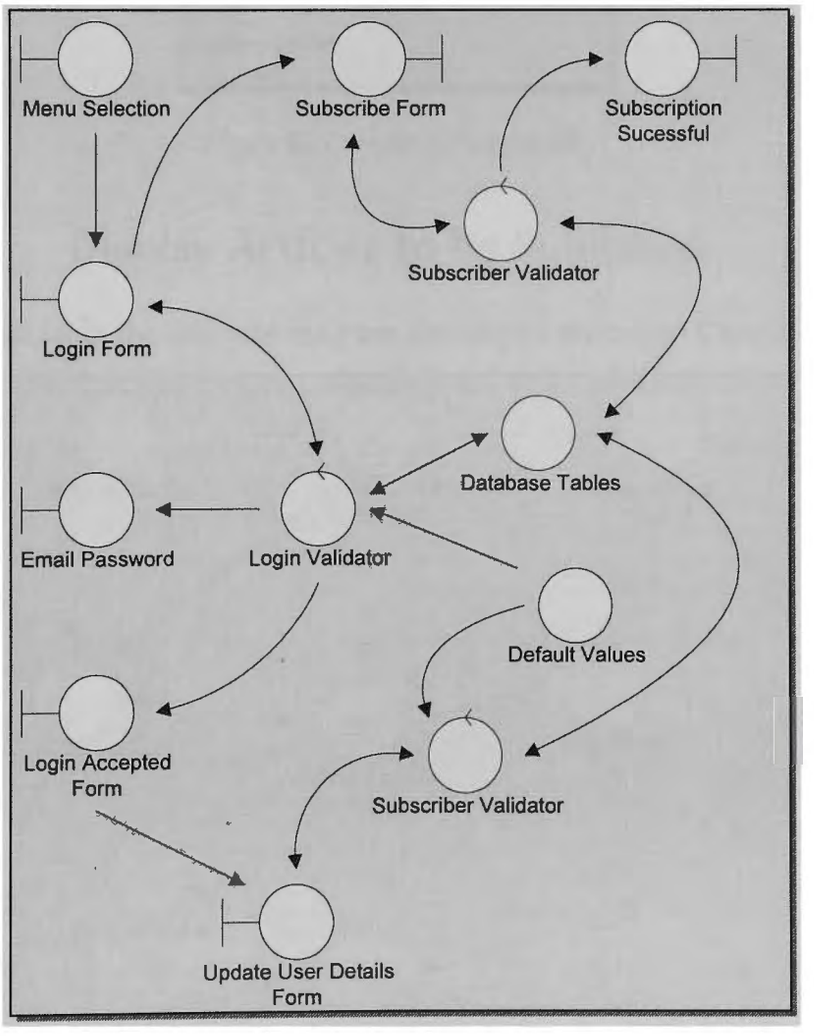


Figure 6. Subscribe/login to JIPAM

4.1.4. Display Current Volume/Issue

Figure 7 below is the analysis model derived from Use Case No. 4.

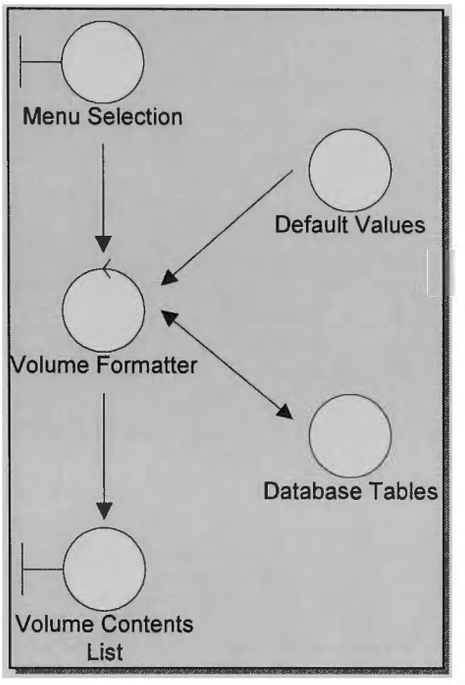


Figure 7. Current Volume/Issue

4.1.5. Display Articles to be Published

Figure 8 below is the analysis diagram developed from Use Case No.5.

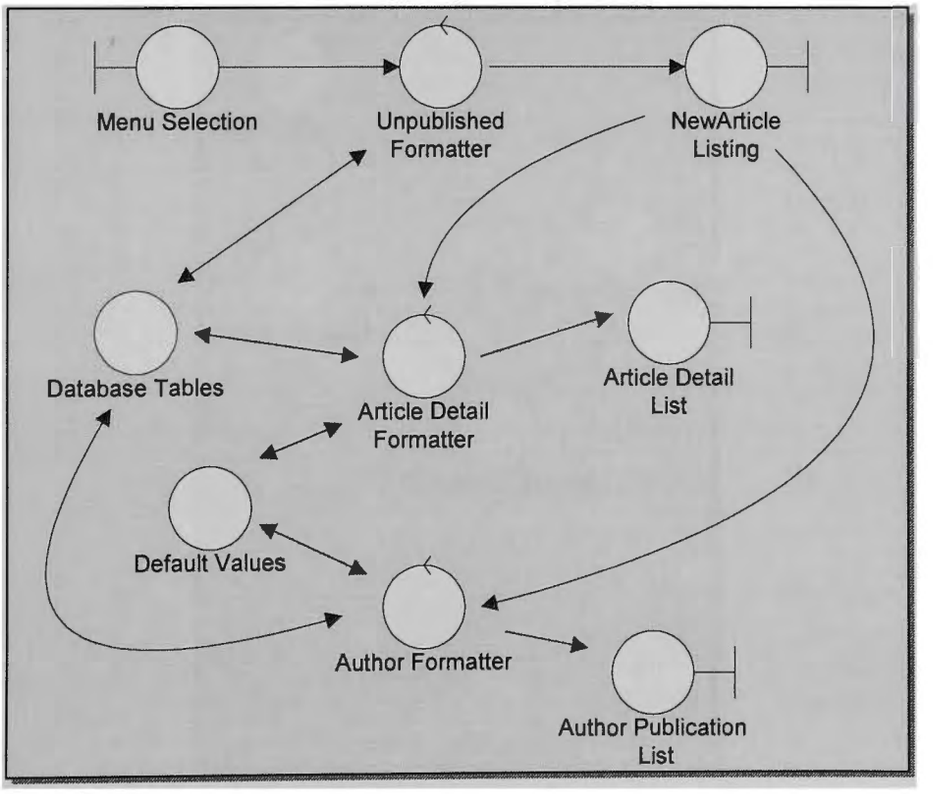


Figure 8. Articles Yet to be Published (New Papers)

4.1.6. Display List of Editors

Figure 9 below is the analysis diagram developed from Use Case No.6.

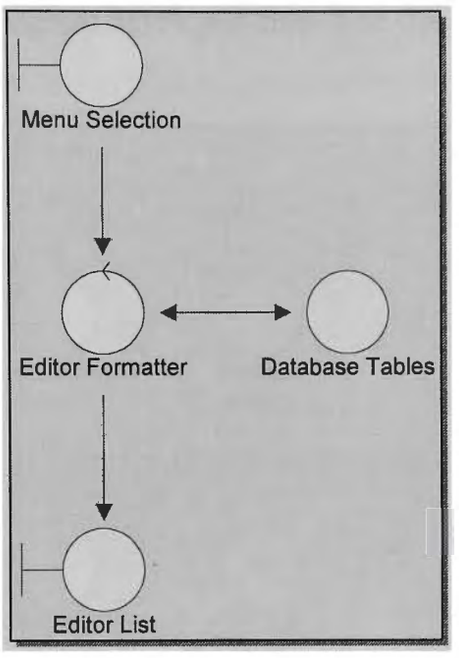


Figure 9. List of Editors

4.1.7. Provide Feedback

Figure 10 below is the analysis diagram developed from Use Case No.7.

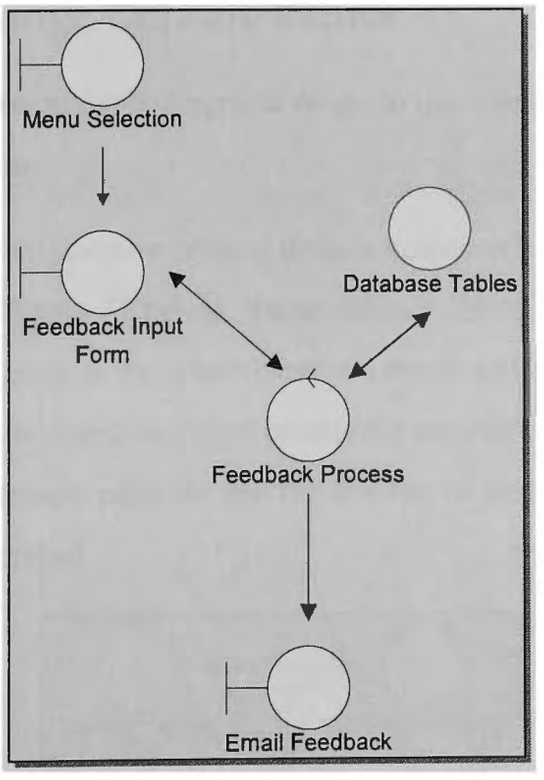


Figure 10. Provide Feedback

4.1.8. Static Pages

There are several existing static pages defined in the requirement model. Figure 11 below is the analysis diagram developed from Use Case No.8 which details all the static pages on the site.

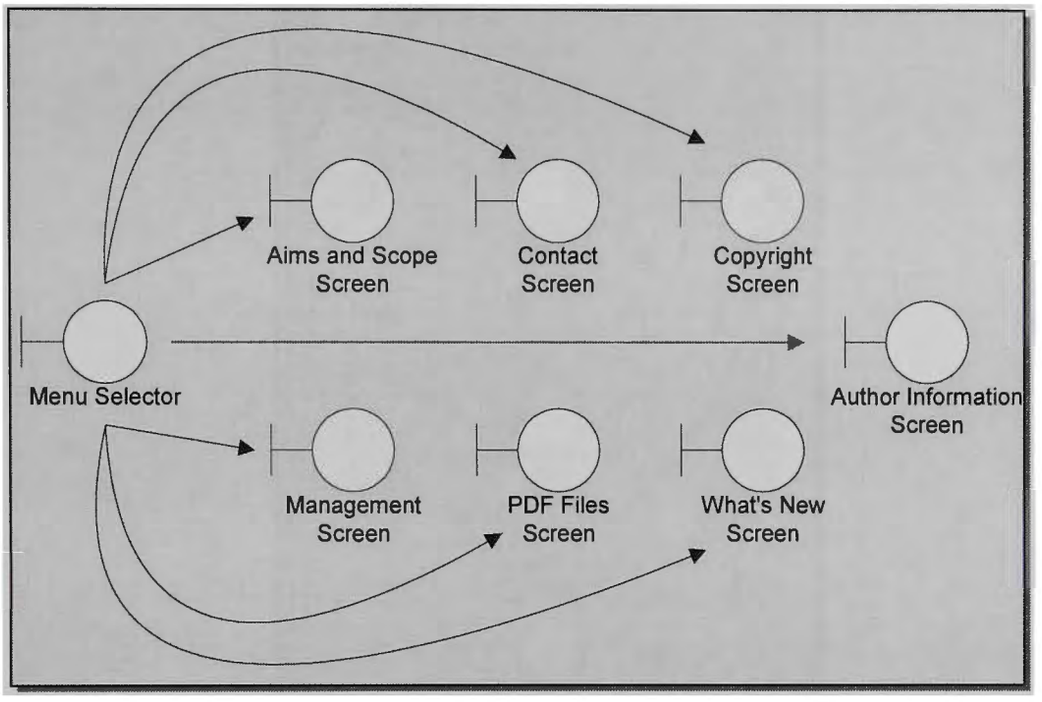


Figure 11. Static Pages

4.1.9. Change Default Values

The remaining seven analysis diagrams relate to use cases where the actor is the JIPAM administrator.

Firstly, the system will require several default values to be maintained by the administrator. See Figure 12 below. These default values relate to Use Case No. 12 and refer to values such as the administrator's email address (for feedback etc). Other default values will include maximum counts for various user activities (e.g. the number of login attempts possible and the number of search results which will be displayed on one screen).

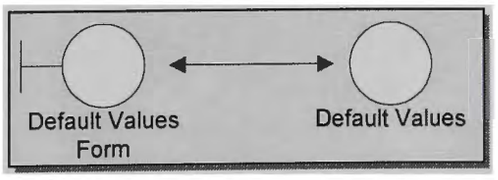


Figure 12. Change Default Values

4.1.10. Create an Article/Change Article Details

The next analysis diagram (Figure 13) represents two use cases (i.e. Use Case No's 13 and 14) where the administrator creates or changes the details about an article .

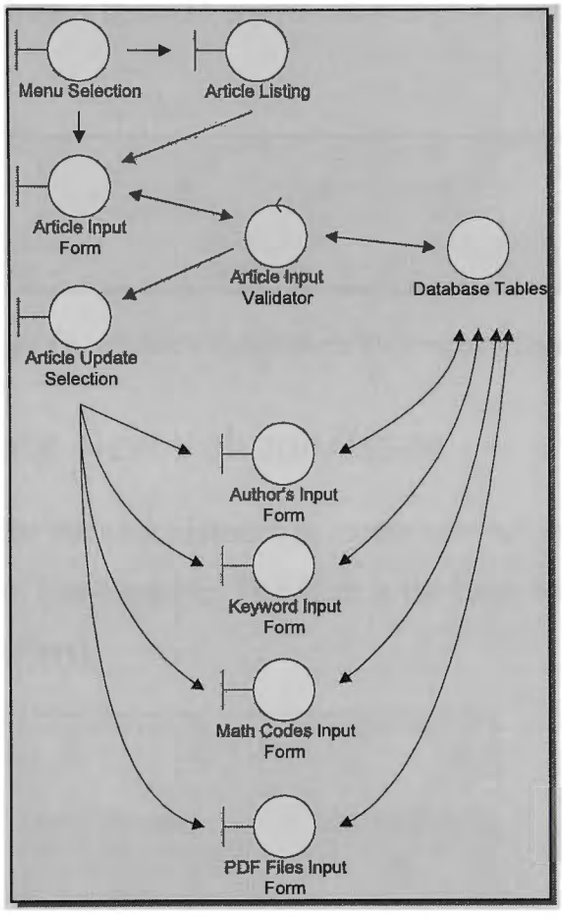


Figure 13. Create/Change an Articles Details

4.1.11. Add/Delete/Change a Subscriber's Details

The next analysis diagram represents three use cases (i.e. Use Case No's 15, 16 and 17) where the administrator may need to add, delete or change a subscriber's details.

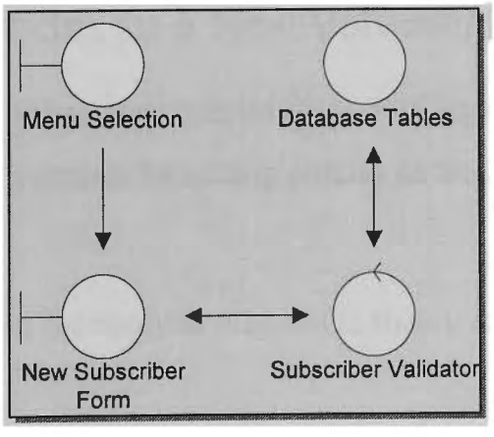


Figure 14. Add/Delete/Change a Subscriber's Details

4.1.12. Update a Subscriber's Subscription Date

To meet the client's requirement to provide the functionality of being able to charge for access to full text articles at some stage in the future, it will be necessary to maintain time based data. Figure 15 shows the analysis diagram relating to Use Case No. 18.

Figure 15. Update a Subscriber's Subscription Date

4.1.13. Create New Volume/Issue

It will be necessary for the administrator to create new volumes/issues of JIPAM on an on-going basis, i.e. Use Case No. 19 which is the basis for the following analysis diagram (Figure 16 refers).

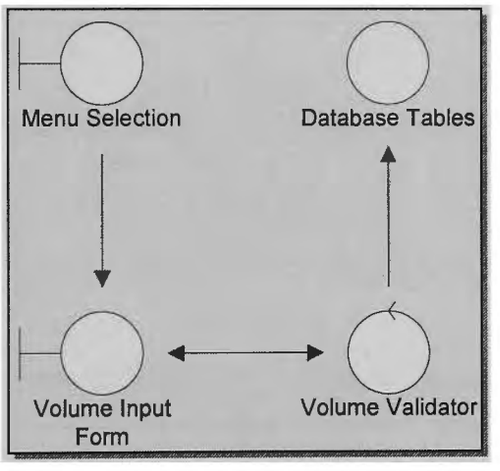


Figure 16. Create a New Volume/Issue

4.1.14. Add Articles to a New Volume/Issue

After the new volume/issue has been created (Figure 16 above), the administrator next builds up its table of contents by adding articles as they are received from the Accepting Editors.

Use Case No. 20 refers and the analysis diagram is shown as Figure 17 on the following page.

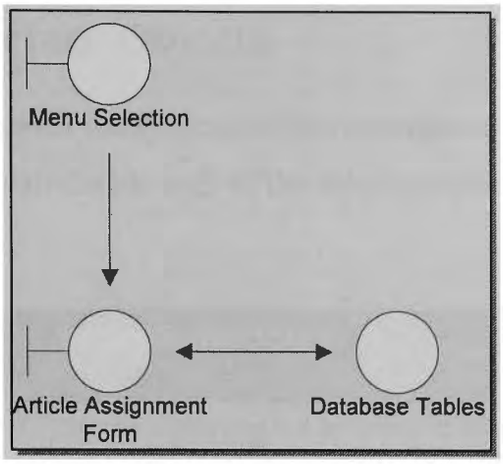


Figure 17. Add Articles to New Volume/Issue

4.1.15. Add/Change an Institution

The JIPAM administrator also needs to be able to add or change details regarding the institutions that general users and subscribers are affiliated with (Use Case No's 22 and 23 refers).

The analysis diagram is shown as Figure 18 below.

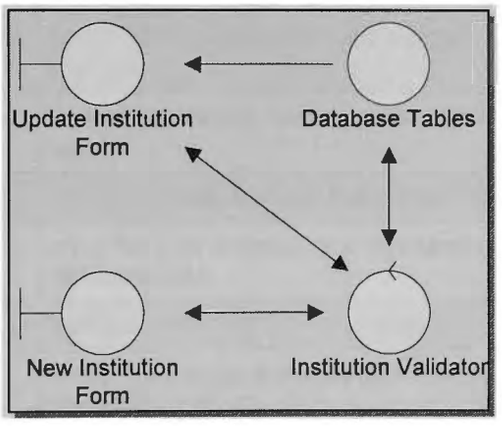


Figure 18. Add/Change an Institution

4.2. OBJECTS AND THEIR RESPONSIBILITIES

A list of all three objects types used in the analysis combined with a brief description of their roles and responsibilities is provided next. As Jacobson et al states, 'we should state explicitly which object is responsible for which behaviour in the use case' (1992, p.175).

4.2.1. Interface Objects

Actors communicate with the system via the interface objects. Details of the responsibilities associated with each of the interface objects is provided in Table 2.

INTERFACE OBJECTS	RESPONSIBILITIES	SUBSYSTEM		
Aims And Scope Screen	Screen displaying the aims and scope of JIPAM	View		
Article Assignment Form	Input form for assigning an article to a volume		Mtce	
Article Details	Screen showing all details about an article	View		
Article Input Form	Input form for accepting data for a new article		Mtce	
Article Listing	Screen displaying the list of articles		Mtce	
Article Titles List	Screen that lists article titles	View		
Article Update Form	Input menu form that provides choices for additional data for an article		Mtce	
Author's Information Screen	Screen displaying the information advising author's about JIPAM publication requirements.	View		
Author's Input Form	Entry form for assigning authors to a article		Mtce	
Author Publication List	Screen listing details about an author as well as all their articles	View		
Contact Screen	Screen for display how to contact JIPAM editorial team	View		
Copyright Screen	Screen for discussing the copyright issues	View		
Default Values Form	Input form for entering and maintaining the system default values		Mtce	
Editor List	Screen displaying the list of editors	View		
Email Feedback	Email form to the administrator concerning feedback	View		
Email Password	Email form to a subscriber advising them of their password	View		
Feedback Input Form	Input form that accepts feedback data	View		
Keyword Input Form	Input form that assigns keywords to an article		Mtce	
Login Accepted Form	Form that indicates that login was successful	View		
Login Form	Form that accepts the login details of subscribers	View		
Management Screen	Screen that will list the members of the JIPAM management board	View		
Math Codes Input Form	Input form that assigns maths codes to an article		Mtce	
Menu Selection	The functionality for providing a choice to the user			Both
New Article Listing	Screen displaying a list of unpublished articles	View		
New Institution Form	Input form for entering data relating to an institution		Mtce	

INTERFACE OBJECTS	RESPONSIBILITIES	SUBSYSTEM		
New Subscriber Form	Administrator's input form for entering all data relating to a subscriber		Mtce	
PDF Files Input Form	Input form that assigns PDF files to an article		Mtce	
PDF Files Screen	Screen which will detail the format of full text articles files	View		
Renew Subscriber Menu	Interface for the administrator to process subscriber's renewal dates		Mtce	
Search Form	Input form that accepts data from a user on which to perform a search	View		
Statistic Report	Report displaying statistic data collected from user's browsers	View		
Subscribe Form	Form that accepts a user's details so they can subscribe	View		
Subscription Successful	Form that advises a user that their subscription was successful	View		
Update Institution Form	Input form for updating data relating to an institution		Mtce	
Update User Details Form	Form a user completes to update their details	View		
Volume Contents List	Screen showing all articles on a published JIPAM volume	View		
Volume Input Form	Input form for entering data relating to a new volume		Mtce	
What's New Screen	Screen which provides the latest information about JIPAM	View		

Table 2. Responsibilities of the Interface Objects

4.2.2. Entity Objects

Entity objects are used to model the information handled by the system over a period of time. Table 3 below lists the JIPAM system entity objects and their responsibilities.

ENTITY OBJECTS	RESPONSIBILITIES	SUBSYSTEM		
Database Tables	Manages persistent data for the system			Both
Default Values	Manages non-database persistent data for the system			Both
Statistic Tables	Manages data collected from users' browsers			Both

Table 3. Responsibilities of the Entity Objects

4.2.3. Control Objects

Control objects contain behaviour which is not naturally placed in either interface or entity objects. Table 4 lists the control objects used in the JIPAM system and their associated responsibilities.

CONTROL OBJECTS	RESPONSIBILITIES	SUBSYSTEM		
Article Detail Formatter	Implementation of the rules to extract and format article details from the database	View		
Article Input Validator	Implementation of the rules that ensure that an article's data is valid		Mtce	
Author Formatter	Implementation of the rules to extract and format an author's details from the database	View		
Editor Formatter	Implementation of the rules that extract editor details from the database	View		
Feedback Process	Implementation of the rules that process feedback data	View		
Institution Validator	Implementation of the rules that ensure an institution's data is valid		Mtce	
Login Validator	Implementation of the rules that govern login to the system	View		
Subscriber Validator	Implementation of the rules that ensure a subscriber's data is valid			Both
Renew All Subscribers	Implementation of the rules governing subscriber's renewal dates		Mtce	
Search Process	Implementation of the rules that perform searches on the database tables	View		
Statistic Formatter	Implementation of the rules that extract data from the statistic tables	View		
Unpublished Formatter	Implementation of the rules to extract and format unpublished article details from the database	View		
Volume Formatter	Implementation of the rules to extract and format a JIPAM volume's details from the database	View		
Volume Validator	Implementation of the rules that ensure the data describing a volume is valid		Mtce	

Table 4. Responsibilities of the Control Objects

4.3. SUBSYSTEMS

Objects derived by the analysis phase may be grouped together into sybsystems to provide a clearer overview of the system. Analysis of the JIPAM Web System

indicated that two subsystems are required. Tables 2, 3 and 4 provide more detailed information about the relationships between the objects and the two subsystems.

4.3.1. Viewing Subsystem

Although these groupings are quite arbitary, Jacobson et al says they should be 'coupled to only one actor, since changes are usually caused by actors' (1992, p.196). In contrast, within the JIPAM system, the first subsystem is derived from the general user and the subscriber actors because they are closely related and the subscriber login function just provides general users with additional access rights (access to the full text articles. 'The division into subsystems should also be based on the functionality of the system' (Jacobson et al, 1992, p.196).

The statistical gathering component has been included in the viewing subsystem because the data collected arises from actions associated with these two groups of actors. Figure 19 below represents a block diagram of the viewing subsystem.

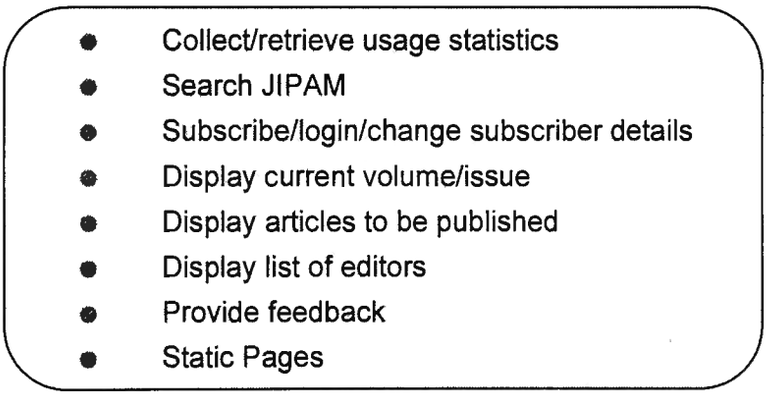


Figure 19. Block Diagram of the Viewing Subsystem

4.3.2. Maintenance Subsystem

The actor, JIPAM Administrator, is the basis of the other subsystem, called maintenance. (see Figure 20 below).

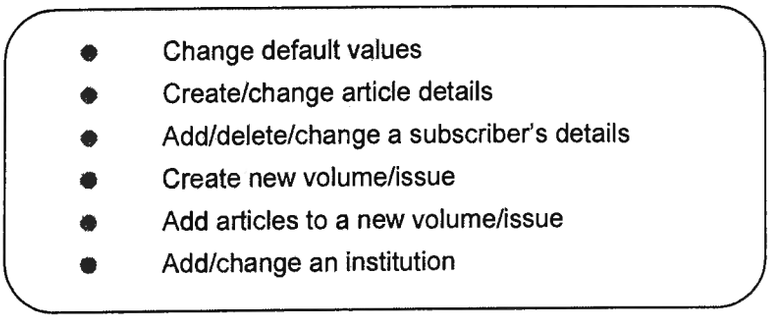


Figure 20. Block Diagram of the Maintenance Subsystem

5. CONSTRUCTION AND DESIGN

According to Jacobson et al, 'the design model will further refine the analysis model in the light of the actual implementation environment'(1992, p.204). This model is used because the ideal behaviour of the analysis model is constrained by operating factors (the real world).

The original intention of this project was to build a web application providing all the additional functionality required of the system. As protoyping was carried out, it was discovered that the ASP software that was specified did not directly support one of the customer's key requirements. Namely, the customer required the full text PDF files to be stored in the database instead of by referencing the files residing on the web server.

There are two possible solutions which allow a file to be loaded into the database. The first involves sending a HTML message to the web server encoded so that the ASP application would communicate with a process running on the client's machine that would return the file directly to the server ASP page. The second method involves embedding the contents of the file in the HTML message. The file is then extracted from the message on the web server.

Neither solution is supported directly by the ASP software and therefore third-party components are required. As these components would have to be purchased (at some considerable cost), it was necessary to find an alternate solution. The solution chosen was to build the administrator subsystem as a stand-alone NT application where the NT application would write directly into the database.

5.1. ANALYSIS MODEL ENTITIES

The analysis phase identified that the JIPAM Web Site: Dynamic Database System could be categorised into two subsystems (Viewing and Maintenance). However, the entity objects have some commonality across both subsystems and are considered in further detail next.

5.1.1. Database Tables

All of the data items needed to support the application excluding statistic data items were extracted from the existing web site. Using standard normalising processes, the database scheme shown in figure 21 was developed.

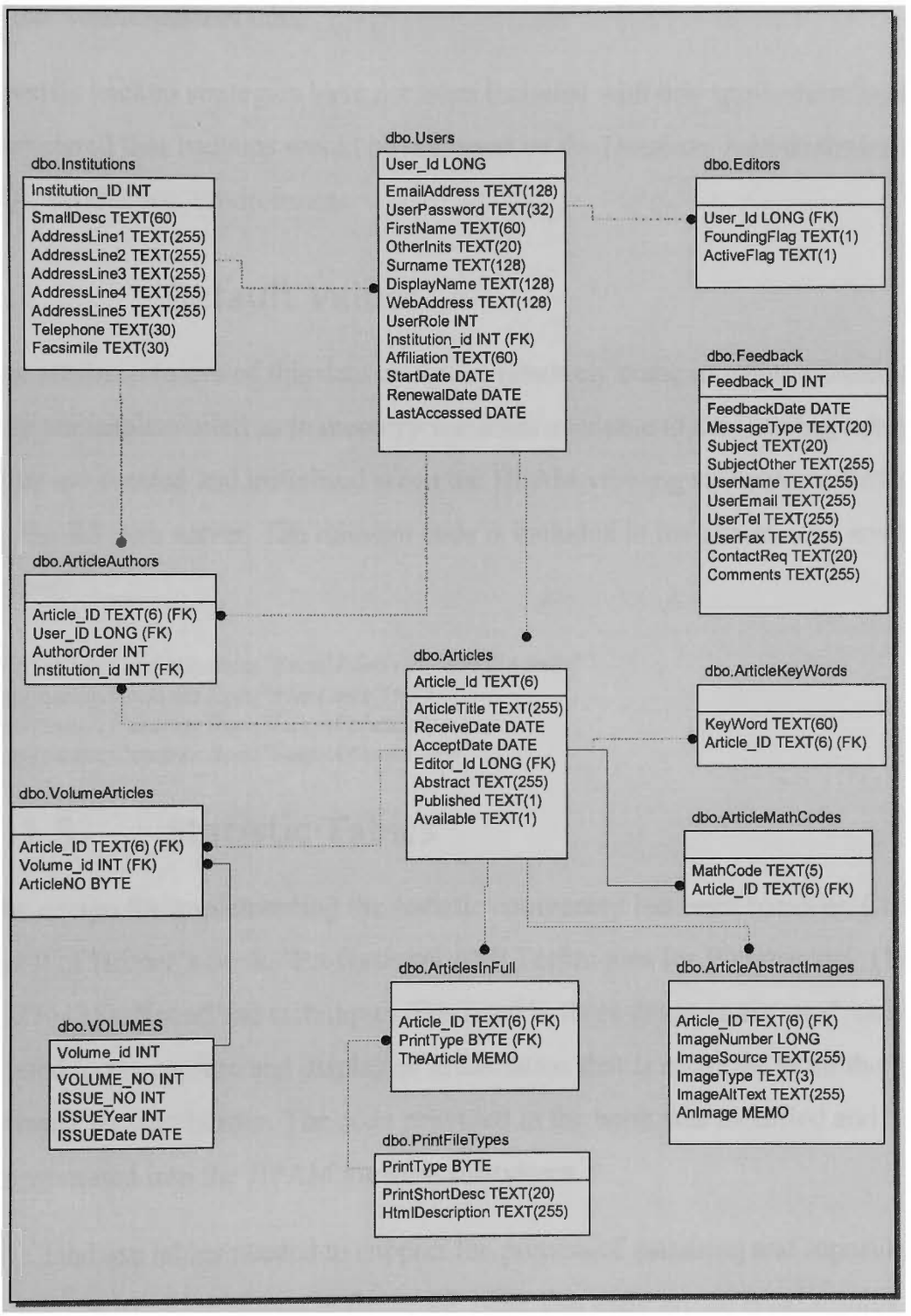


Figure 21. Database Scheme

The only tricky relationship in the displayed scheme is that between an institution_id and an author-article combination. When an author publishes an article, the author does so as member of an particular institution. At some point later in their academic career they may move to another institution and publish additional papers. The first relationship needs to maintained forever and is why the field institution_id is stored in the ArticleAuthors table.

Specific backup strategies have not been included with this application because it is considered that backups would be managed by the Database Administrator in an SQL Production Environment.

5.1.2. Default Values

The attribute values of this data entity are relatively static in value. Consequently, they are implemented as in memory variables available to the viewing subsystem. They are created and initialised when the JIPAM viewing subsytem is first accessed by the IIS web server. The relevent code is included in the global.ASA module (see below).

```
Application.Contents.Item("EmailAdmin")="EmailAdmin"
Application.Contents.Item("MaxLines")=30
Application.Contents.Item("LatestVolumId")=1
Application.Contents.Item("LoginAttemps")=4
```

5.1.3. Statistic Tables

The design for implementing the statistic component has been based on Chapters 8 and 9 of Homer's book, 'Professional ASP Techniques for Webmasters' (1998, p327–425). Not all the techniques discussed in these chapters are used, only those related to the capture and display of information that is extracted from the web browser request header. The code provided in the book was modified and incorporated into the JIPAM viewing subsystem.

The database tables needed to support the process of gathering and reporting on visitor statistics are shown in figure 22. Note that there are no relationships between any of the tables as they are populated under the control of a SQL batch job.

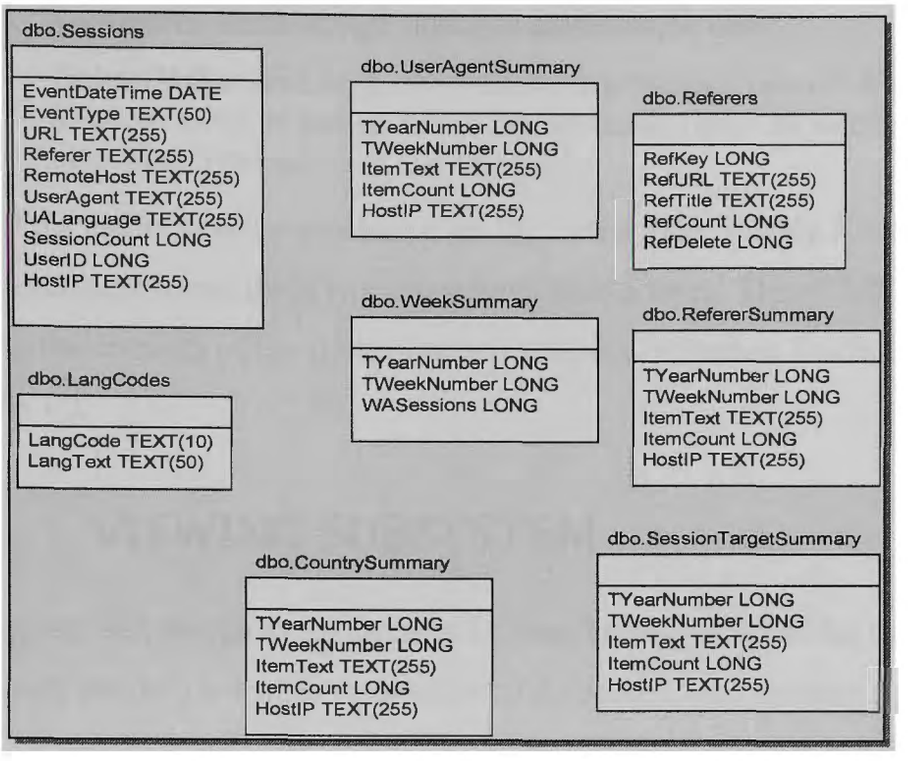


Figure 22. Usage Statistics Database Tables

Every time a user accesses the JIPAM web site, a new session object is created by the web server provided they do not already have a current session object. As part of the creation of the session object, a record is written to the Sessions database table. The code for writing the record is contained in the session's on_start procedure that resides in the global.ASA file in the root JIPAM directory and is as follows.

```
strSQL = "INSERT INTO Sessions " _
      &(EventDateTime,EventType,URL,Referer,RemoteHost,UserAgent
       ,UALanguage,UserID,HostIP)" _
      & " VALUES (GetDate(),'New Session', '" _
      & Request.ServerVariables("URL") & "', '" _
      & Request.ServerVariables("HTTP_REFERER") & "', '" _
      & Request.ServerVariables("REMOTE_HOST") & "', '" _
      & Request.ServerVariables("HTTP_USER_AGENT") & "', '" _
      & Request.ServerVariables("HTTP_ACCEPT_LANGUAGE") & "', " _
      & CLng(Session.SessionID) & ", '" _
      & Request.ServerVariables("LOCAL_ADDR") & "')"
  WriteSessionData strSQL
```

The record written to the database contains information extracted from the users' request header and it is raw data. The raw data needs to be summarised into other tables before it can be analysed and reported on. The following two database procedures are provided to support this summarisation process. The actual code is shown as Appendix B.

- ***SummariseSession.SQL*** which summaries the data
- ***DeleteOldRecords.SQL*** removes the summarised records from the Sessions table as well as removing all records over 26 weeks old from the other statistical tables in the database.

As part of the database setup procedure, an SQL job called Weekly JIPAM Session Update2 is created to run these two procedures once a week. The HTML code for displaying the contents of the summarisation tables is in module TrafficReports.ASP (Appendix C).

5.2. VIEWING SUBSYSTEM

This subsytem was design to run from an Internet browser. One of the customer's requirements was to use reduce the amount of Javascript used as much as possible because a number of their customers still had primitive browsers that did not support its use.

The requirement to support primative browsers also had an affect on the menu design of the HTML pages. As HTML tables are supported in nearly all browsers, it was decided to use HTML tables in preference to HTML frames. Each HTML page is assigned a name and the menu that is displayed to a user depends on both the page name and the user's role in the system. In Phase 1, only the Administrator role has an addition menu selection.

5.2.1. Final Design

In the final design, each page returned from the IIS server is constructed from the combination of four files. The main file is the file shown as being the body of the HTML page in the figure. This file uses the IIS facility of Server-Side includes to incorporate the three files that incorporate the HTML forming the document header, the menu sidebar and the document footer.

This approach was chosen for the following two reasons.

- To promote a common look to all JIPAM web pages
- To reduce future maintenance needs by incorporating all common look facilities in one place.

The final design for the viewing subsystem is shown in Figure 23.

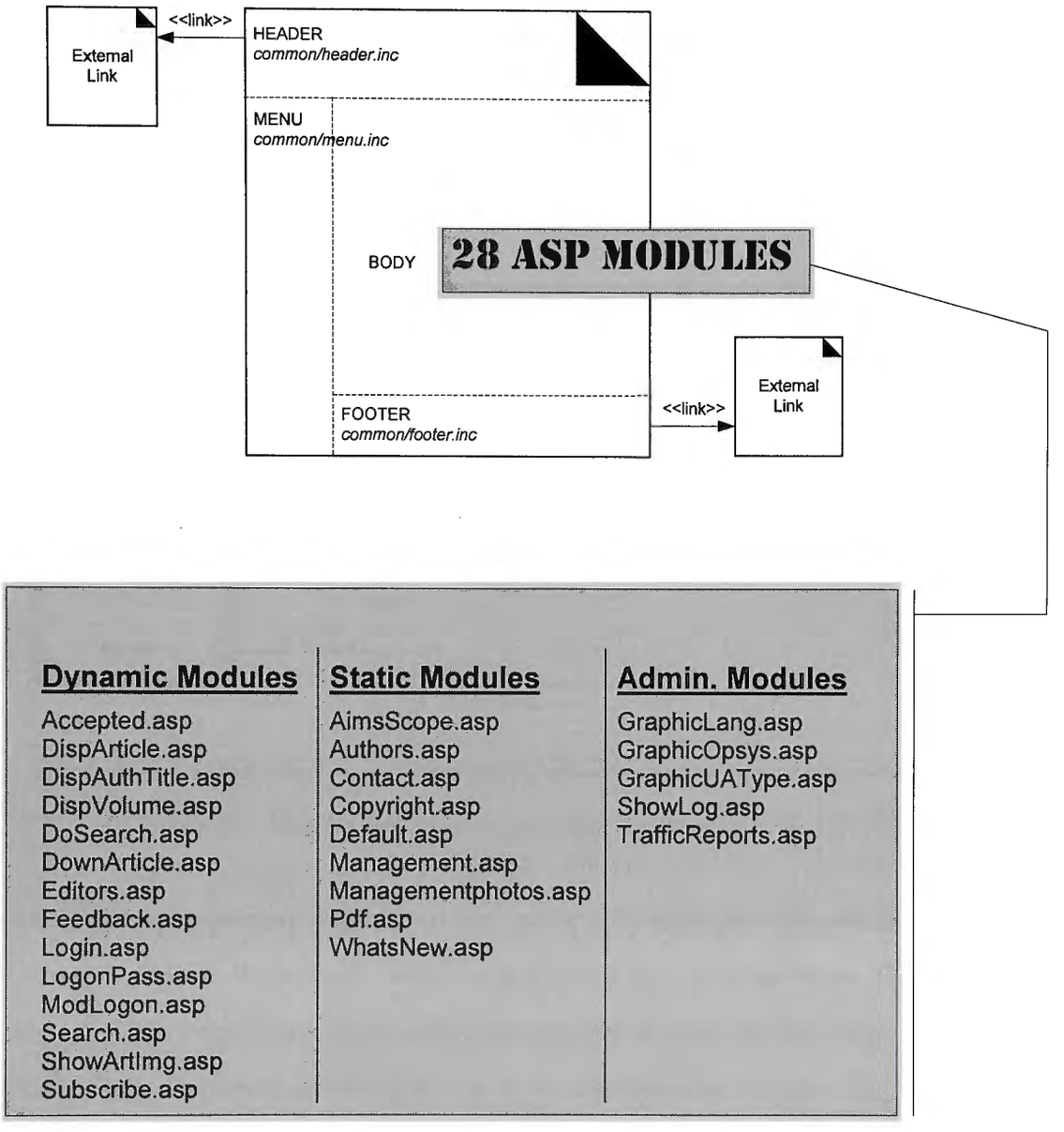


Figure 23. Viewing Subsystem Final Design

The HTML code produced by each of the modules listed in figure 23 is placed inside a HTML data cell. The <TD> and </TD> HTML data cell pair are output as the last statement of file common/menu.inc and the first statement of file common/footer.INC.

The relationship between the buttons on the menu bar and the ASP modules is shown in Figure 24 (shown on the following page).

The menu facility on each web page in Phase 1 of the JIPAM web system uses buttons to control the actions that can be performed by users. The same menu items

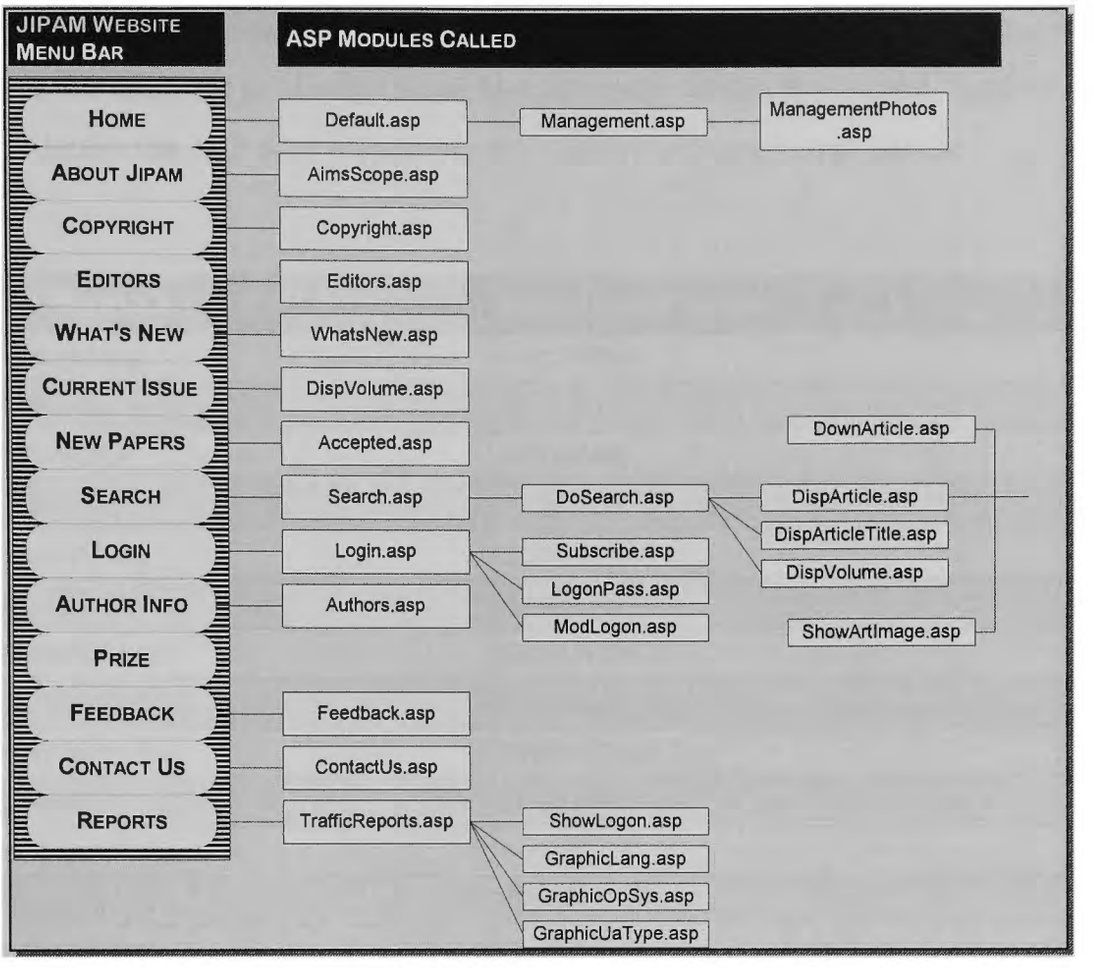


Figure 24. Relationship Between the Menu Structure and the ASP Pages

are provided as hyperlink markers in the footer. The state of both menus is controlled by the asp variable, PageName which is initialised by each asp page. The value set for the variable PageName determines whether the button for that page is displayed as being up or displayed as being down. It also determines whether the corresponding hyperlink marker in the footer is turned on or off.

As an example, the ASP code for displaying the list of editors model is shown below.

```
<%  if PageName="EDITORS" then %>
SRC="/jipam/images/Editors-down.gif"  > </TD>
<%  else  %>
SRC="/jipam/images/Editors-up.gif" USEMAP="#FPEditors" > </TD>
<%  end if  %>
```

The corresponding code in the footer is shown next.

```
if PageName<>"EDITORS" then%>
<NOBR>[ <A HREF="Editors.asp">Editors </A>]</NOBR>
<% else %>
<NOBR>[ Editors ]</NOBR>
<%  end if
```

Note that the ASP module names are the external file name whereas the menu page name is the value set to identity those files internally within the system. Table 5 below details the ASP files names and their associated menu page names.

ASP MODULE (FILE NAMES)	MENU PAGE NAME
Accepted.asp	ACCEPTED
AimsScope.asp	SCOPE
Authors.asp	AUTHORS
Contact.asp	CONTACT
Copyright.asp	COPYRIGHT
Default.asp	MAIN
DispArticle.asp	DISPLAYART
DispAuthTitle.asp	DISPLAYAUTH
DispVolume.asp	DISPLAYVOL
DoSearch.asp	DOSEARCH
DownArticle.asp	noname
Editors.asp	EDITORS
Feedback.asp	FEEDBACK
GraphicLang.asp	TRAFFICLAN
GraphicOpSys.asp	TRAFFICOPS
GraphicUaType.asp	TRAFFICUAT
Login.asp	LOGON
LogonPass.asp	LOGON
Management.asp	MANAGEMENT
ManagementPhotos.asp	MANAGEMENT
ModLogon.asp	MODLOGON
Pdf.asp	PDF
Search.asp	SEARCH
ShowArtImg.asp	NONAME
ShowLog.asp	TRAFFICLOG
Subscribe.asp	SUBSCRIBE
TrafficReports.asp	TRAFFIC
WhatsNew.asp	WHATSNEW

Table 5. Association Between ASP Page Names and Menu Page Names

The ASP modules with a page name of 'noname' in the table are not ASP pages called by the menu system. These 'noname' pages provide support to other ASP pages to complete actions that enable those calling pages to be completed correctly.

5.2.2. Security

Security is one of the major issues in using a database in the web environment. In order to restrict the access that general users have, a separate database user (INTERNET) is created in the setup process for the JIPAM application. This user is only given execute rights to the database and therefore can only execute database procedures for which they have been given permission. This means that, even if an Internet user found out the userid and password of the created database user, they could not see what tables exist in the database or the structure of any database table.

A further security feature involves the use of the Session object provided in IIS. All details relating a general user/subscriber including their login status are kept on the web server. While this increased the memory used by individual users, it has meant that messages to and from the web server were reduced in size. It has also meant that user information would not be returned to a user, thereby preventing possible fraudulent activities.

The initialization of the data items relating to security occurs when the Session object is created for a user. The creation of a Session object for individual JIPAM users occurs whenever a JIPAM web page is accessed. The variables initial code is in the Session on_start procedure which is contained in the global.ASA file.

5.2.3. Interface Objects

The viewing subsystem interface objects and the ASP modules on which they are implemented is show in Table 6. An ASP module may implement more than one interface object.

INTERFACE OBJECT	ASP MODULE	RELATED SQL PROCEDURE
Aims And Scope Screen	AimsScope.asp	
Article Details	DispArticle.asp	

INTERFACE OBJECT	ASP MODULE	RELATED SQL PROCEDURE
Article Titles List	Dosearch.asp	
Author's Information Screen	Authors.asp	
Author Publication List	DispAuthTitle.asp	
Contact Screen	ContactUs.asp	
Copyright Screen	Copyright.asp	
Editor List	Editors.asp	
Email Feedback	Feedback.asp	
Email Password	LogonPass.asp	
Feedback Input Form	Feedback.asp	
Login Accepted Form	Login.asp	
Login Form	Login.asp	
Management Screen	Management.asp	
New Article Listing	Accepted.asp	
PDF Files Screen	Pdf.asp	
Search Form	Search.asp	
Statistic Report	TrafficReport.asp	
Subscribe Form	Subscribe.asp	
Subscription Successful	Subscribe.asp	AddUserFull
Update User Details Form	ModLogon.asp	ModUserLong, GetUserDetails
Volume Contents List	DispVolume.asp	DispVolArticles, VolumeList
What's New Screen	WhatsNew.asp	

Table 6. Viewing Subsystem - Interface Objects

5.2.4. Control Objects

The ASP pages that implement the control objects are shown in Table 7. Also shown are the SQL database procedures that assist in implementing the control objects.

CONTROL OBJECTS	ASP MODULE	RELATED SQL PROCEDURE
Article Detail Formatter	DispArticle.asp	DispArtDetails DispArtAuthors DispAuthor DispArtKeywords DispArtMathCodes DispArtDownloads
Author Formatter	DispAuthTitle.asp	DispAuthor DispAuthArtList
Editor Formatter	Editors.asp	FullEditorsList
Feedback Process	Feedback.asp	AddFeedback

CONTROL OBJECTS	ASP MODULE	RELATED SQL PROCEDURE
Login Validator	Login.asp	ValidateUser
Search Process	DoSearch.asp	SearchAbstracts SearchAuthors SearchKeywords SearchMathCodes SearchTitles
Statistic Formatter	ShowLog.asp GraphicLang.asp GraphicOpSys.asp GraphicUaType.asp	
Unpublished Formatter	Accepted.asp	UnPubArticles
Volume Formatter	DispVolume.asp	DispVolArticles; VolumeList

Table 7. Viewing Subsystem - Control Objects

Where there is no entry in the *Related SQL Procedures* column in Table 7 above, the database access was constructed directly in SQL using code provided within Chapters 8 and 9 of Homer's book (1998).

5.3. MAINTENANCE SUBSYSTEM

The maintenance program was developed to run as a stand-alone NT program and it was written in the C++ language using the Borland Builder Program. The development environment provided by this program is very similar to Microsoft Visual Basic environment. An application program is developed in this environment using pre-supplied components that are inserted into the containers that provide the underlying structure. The entire program is built using these components and C++ objects that are developed to support the user's application.

5.3.1. Administration Program Users Guide

Jacobson et al says that 'documentation of the system should be produced by the developers' but that 'manuals, both for maintenance and for users, should be written by people with special skills for this' (1992, p.462). Their views on documentation have been followed as guiding principles for the maintenance subsystem, however, significiant effort has been made to develop effective maintenance documentation.

Thus, a detailed manual titled 'JIPAM Administration Program Guide' has been produced. It is structured into seven chapters as listed below.

- Chapter 1 Introduction
- Chapter 2 Admin.Program Installation
- Chapter 3 Subscribers
- Chapter 4 Articles
- Chapter 5 Volumes/Issues
- Chapter 6 Institutions
- Chapter 7 Statistics

Although normally this guide would be bound and presented as a separate document for ease of use by the administrator, it has been collated within this thesis as Appendix E.

5.3.2. Database Issues

The database objects provided in the environment, for example classes based on TTable, TQuery and TDBGrid make it a simple matter to view and scroll through the database tables so this approach was chosen as the underlying methodology for accessing individual records in the database. These database objects connect to the database using the ODBC interface.

In addition, a single variable was created for every field in the database and it controls the size of any input field associated with the associated database field. This strategy was used to reduce future maintenance effort if the size of the database fields ever need to be changed. The variable definitions appear in the module that contain the form on which the associated field is input.

A number of issues arose while developing the maintenance subsytem. The first issue related to the database viewing grid that formed the starting point for all activities. When an individual record was modified in a dialog box, the changed record was not displayed with the updated details in the viewing grid. This was eventually traced to the fact that the tables were relational tables accessed through the ODBC pipeline and they were cached in memory as they were retrieved in the underlying query. Despite researching as many different ways as possible to overcome this issue including making the underlying query supporting the viewing grid, the issue

remains. It is only overcome by reissuing the query against the database. All viewing grids support filter and sort operations which create a new query on the underlying database table when they are changed.

The second issue concerned the size of variable length text fields, in particular the article abstract field. Despite the abstract field being created with a 6,144 variable character size, the viewing grid and the memo edit box saw the field as a 255 character field. The problem was eventually traced to the ODBC driver. Even when accessing the abstract field directly through the ODBC driver, only the first 255 characters were being accessed. The solution was to cast the abstract field as text in the query that retrieved the abstract field from the database. This resulted in the correct 'type' being returned which made the ODBC driver behave correctly.

The third issue was caused by the Borland database driver. This driver sits above the ODBC API and acts as an interface to their database objects, e.g. their database table object (TTable). Because the PDF files were going to be stored in the database as BLOBS (Binary Large ObjectS) and the Borland database objects supplied a BLOB object, the writing of the PDF files to the database should have been easy.

This was not the case.

The Borland database driver was truncating the PDF file at the size defined by the 'BLOB SIZE' parameter in their database administrator. Despite their documentation which suggested that there should be no problem with using their BLOB object, the only solution found was to write the PDF files directly to the database using the ODBC API.

Because of these issues, there were a number of database activities which were not implemented as database procedures. Instead, some were implemented as straight SQL code.

5.3.3. Final Design

The relationship between the main menu items in the maintenance program and the C++ modules is shown in Figure 25 on the following page. The main menu items are

implemented as push buttons on the main program form. When selected, each of the buttons brings up a screen showing the contents of the underlying table.

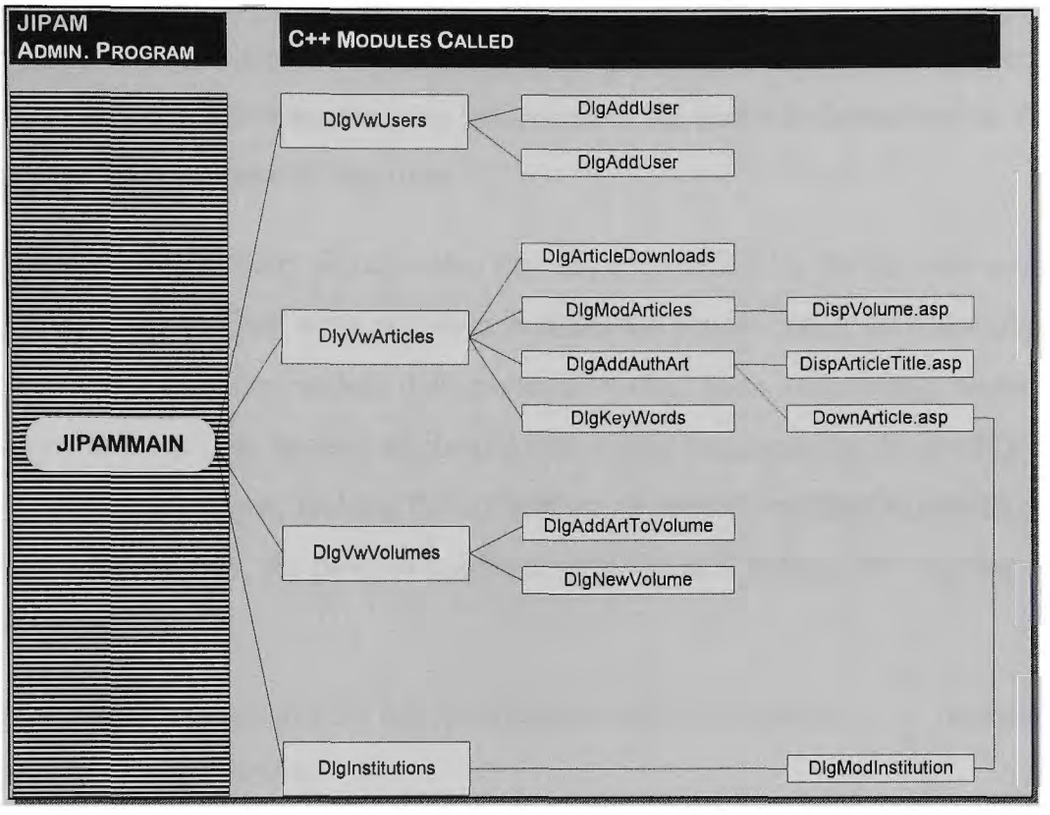


Figure 25. Maintenance Subsystem Final Design

5.3.4. Security

Additional security (other than that provided by the NT operating system) has not been built into the maintenance program. As the program needs to be installed for each individual user, they will have rights to use and execute the program. The administration program installation process is fully detailed in the Administration Program Guide (Appendix E).

During the setup of the database (see Appendix B - SetUp.SQL which provides the relevant code), a database login id (JPADMN with password JPADMN) is created for the user of the maintenance subsystem. This login id has administration rights to the database.

5.3.5. Interface Objects

The implementation of all interface objects was completed using dialog box forms. In most cases, all use cases had their own dialog box but some interfaces were implemented using the same dialog box. This approach was chosen to reduce the amount of code required to implement the display aspects of the dialog box. Where the same dialog box was used, the behaviour of the box was dependent on the interface being shown at that time.

A number of interfaces objects other that those specified by the use case analysis were developed. They were provided to make the process more efficient. An example of this is the module that creates an author from a subscriber when adding authors to an article. Instead of closing that dialog box, opening the modify subscriber dialog box, making the subscriber an author, and then re opening the original dialog box, the process can be done while still holding the original dialog box open.

The maintenance subsystem interface objects and their relevant C++ modules are shown in Table 8 below.

INTERFACE OBJECT	C++ MODULE
Article Assignment Form	DlgAddArtToVolume
Article Input Form	DlgModArticles
Article Update Form	DlgModArticles
Author's Input Form	DlgAddAuthArt
Keyword Input Form	DlgKeyWords
Math Codes Input Form	DlgKeyWords
New Institution Form	DlgModInstitution
New Subscriber Form	DlgAddUser
PDF Files Input Form	DlgArticleDownloads
Update Institution Form	DlgModInstitution
Volume Input Form	DlgNewVolume

Table 8. Maintenance Subsystem Interface Objects

The interface item 'Renew Subscriber Menu 'has been implemented in this version as SQL database job 'Weekly JIPAM Session Update2'. This job automatically runs each week summarising the Sessions database table into the other statistic tables.

5.3.6. Control Objects

The implementation of the control objects and their associated C++ modules in the maintenance subsystem are shown in the following table (Table 9).

CONTROL OBJECT	C++ MODULE
Article Input Validator	DlgModArticles
Institution Validator	DlgModInstitution
Subscriber Validator	DlgAddUser
Volume Validator	DlgNewVolume

Table 9. Maintenance Subsystem Control Objects

5.4. TESTING

Testing was undertaken in parallel through all stages of construction to ensure that the use cases were successfully completed.

As Jacobson et al specified, 'each use case is initially tested separately, from an external viewpoint. These tests are thus based on the requirements model. When all use cases have been tested separately, the entire system is tested as a whole' (1992, p.333).

6. SUMMARY

The research problem involved the redevelopment of the existing JIPAM Web Site from a site that was based on static content pages to one that was based on database technology. In addition, it was a client requirement that the new web site was to be based on Active Server Pages technology delivered from an SQL7 database.

In arriving at a solution to this problem, it was necessary to investigate the technologies proposed by the client to explore both how they should be used as well as their limitations. Discussions on these issues were covered in Chapter 2.

The methodology chosen to arrive at a software solution was the use case driven approach of Jacobson et al (1992). Discussions on this methodology and how the methodology applied to the solution were covered in Chapter 3 through to Chapter 5.

All of the research objectives set out in the introduction were achieved. A dynamic web site has been created for the JIPAM Managerial Board. It is based on a database framework and it uses an ASP front-end to provide customers with dynamic access to the journal articles held in the database. It also solves a number of problems that affected the existing JIPAM site.

In summary, the software solution which is being delivered ...

- reduces the maintenance effort by automatically providing links to other documents
- provides a search facility for visitors so that they can query articles on a number of criteria fields
- restricts access to the full text articles to only those visitors that subscribe and subsequently logon to the site

It did not however solve the problem of tracking the peer review process of articles nor did the subscription process include charging for site access.

After discussions with my academic supervisor, it was decided that the research project should focus on developing an extensive and effective foundational platform to meet the basic requirements.

The facility to track the review process and the facility to charge a fee to access are therefore left to those responsible for the next generation of the JIPAM Web Site.

Another potential future enhancement involves the provision of a more complex search engine to query the database. In phase 1 of the JIPAM Web Site development process, a simple search process is provided with the search being performed on a single field at a time. As the JIPAM web site matures, it may be beneficial to allow logical searching amongst multiple fields.

REFERENCES

Bandyopadhyay, A. 1999, *'Accessing Sci-Tech Literature: Commercial Document Delivery Services and Online Full-Text Databases'*, Collection Building, Vol. 18, Issue 1, pp. 10–15.

Conallen, J. 1999, 'Modeling Web Application Architectures with UML', Association for Computing Machinery, *Communications of the ACM,* New York, Oct, pp.64–70. Full-text [online]. ProQuest, Bell & Howard, [Accessed 28/11/00].

Feiler, J. 1999, *Database-driven Web Sites,* Morgan Kaufmann, San Francisco.

Fletcher, L.A. 1999, *'Developing an Integrated Approach to Electronic Publishing: Tailoring your Content for the Web'*[online], Learned Publishing, Vol.12, No.2, April, 107–118. Available http://www.catchword.com/alpsp/09531513/v12n2/contp1-2.htm accessed 10/9/00.

Fowler, M. nd, *'Techniques for Object Oriented Analysis and Design'* [online]. Available http://www2.awl.com/cseng/titles/0-201-89542-0/techniques/useCase.htm accessed 12/11/00.

Gadd, N. 2000, *'E-Journal Slashes Publication Time, Costs'*, Nexus, Vol.10, No.8, October, p.2, Victoria University.

Homer, A. 1998, *Professional ASP Techniques for Webmasters,* Wrox Press Ltd., Birmingham.

Jacobson, I., Christerson, M., Jonsson, P. & Övergaard, G. 1992, *Object-Oriented Software Engineering: A Use Case Driven Approach,* Addison-Wesley, Harlow, England.

Keyton Weissinger, A. 1999, *ASP in a Nutshell: A Desktop Quick Reference,* O'Reilly & Associates, Inc. Beijing.

Kobryn, C. 1999, 'UML 2001: A Standardization Odyssey', *Communication of the ACM,* Vol.42, No.10, October, pp 29–37.

Valaparty, K. 2000, 'Use Case-Basics', *Journal of Object-Oriented Programming,* Vol.12, No.9, February, pp 4–8. Full-text [online].

Weintraub, I. 2000, *'The Impact of Alternative Presses on Scientific Communication',* The International Journal on Grey Literature, Vol.1, Issue 2, pp. 54–59.

RFC 1945

RFC 2068

RFC 1867

Appendix A

JIPAM WEB SITE
Dynamic Database System

Installation CD-ROM

INSTALL CD IS UNAVAILABLE

—

INTELLECTUAL PROPERTY RIGHTS APPLY

Appendix B

JIPAM WEB SITE
Dynamic Database System

SQL Server 7 Programming Code

Appendix B

MODULE NAME – SETUP.SQL

```
USE JIPAM
SET DATEFORMAT dmy
/****** Object:  User internet    ******/
/*  remove user so login can be changed */
if exists (select * from sysusers where name = N'internet' and uid < 16382)
  BEGIN
  EXEC sp_revokedbaccess N'internet'
  END
GO


/****** Object:  Login internet    ******/
BEGIN
  if exists (select * from master..syslogins where name = N'internet')
   Begin
     EXEC sp_dropuser N'internet'
     EXEC sp_droplogin N'internet'
   end

  BEGIN
    exec sp_addlogin N'internet', N'internet', JIPAM, 'us_english'
   EXEC sp_grantdbaccess N'internet', N'internet'
  END
END
GO

/****** Object:  User JPADMN    ******/
/*  remove user so login can be changed */
if exists (select * from sysusers where name = N'JPADMN' and uid < 16382)
  BEGIN
  EXEC sp_revokedbaccess N'JPADMN'
  END
GO


/****** Object:  Login JPADMN    ******/
BEGIN
  if exists (select * from master..syslogins where name = N'JPADMN')
   Begin
     EXEC sp_dropuser N'JPADMN'
     EXEC sp_droplogin N'JPADMN'
   end

  BEGIN
     exec sp_addlogin N'JPADMN', N'JPADMN', 'JIPAM', 'us_english'
     EXEC sp_grantdbaccess N'JPADMN', N'JPADMN'
     exec sp_addrolemember 'db_owner', 'JPADMN'
  END
END
GO

/****** Object:  User JipamLog    ******/
/*  remove user so login can be changed */
if exists (select * from sysusers where name = N'JipamLog' and uid < 16382)
  BEGIN
  EXEC sp_revokedbaccess N'JipamLog'
  END
GO

/****** Object:  Login JipamLog******/
BEGIN
  if exists (select * from master..syslogins where name = N'JipamLog')
  BEGIN
    EXEC sp_dropuser N'JipamLog'
    EXEC sp_droplogin N'JipamLog'
  END
  exec sp_addlogin N'JipamLog',N'JipamLog',N'Jipam',N'us_english'
  EXEC sp_grantdbaccess N'JipamLog', N'JipamLog'
END
GO

/****** Delete all existing tables if present ******/

/****** Object:  Table Sessions    ******/
if exists (select * from sysobjects where id = object_id('dbo.Sessions') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table dbo.Sessions
GO

/******* */
if exists (select * from sysobjects where id = object_id('dbo.ArticleAbstractImages')
and sysstat & 0xf = 3)
   drop table dbo.ArticleAbstractImages
GO

/******* del ArticleKeyWords before Articles */
if exists (select * from sysobjects where id = object_id('dbo.ArticleKeyWords') and
sysstat & 0xf = 3)
   drop table dbo.ArticleKeyWords
GO

/******* del VolumeArticles before Articles,Volumes */
if exists (select * from sysobjects where id = object_id('dbo.VolumeArticles') and
sysstat & 0xf = 3)
   drop table dbo.VolumeArticles
GO

/******* del ArticlesInFull before Articles,PrintFileTypes */
if exists (select * from sysobjects where id = object_id('dbo.ArticlesInFull') and
sysstat & 0xf = 3)
   drop table dbo.ArticlesInFull
GO

/******* del ArticleMathCodes before Articles */
if exists (select * from sysobjects where id = object_id('dbo.ArticleMathCodes') and
sysstat & 0xf = 3)
   drop table dbo.ArticleMathCodes
GO


/******* del ArticleAuthors before Users,Articles */
if exists (select * from sysobjects where id = object_id('dbo.ArticleAuthors') and
sysstat & 0xf = 3)
   drop table dbo.ArticleAuthors
GO

/******* del Authors before Users,Institutions */
if exists (select * from sysobjects where id = object_id('dbo.Authors') and sysstat &
0xf = 3)
   drop table dbo.Authors
GO

/******* del Articles before User_id */
```

```
if exists (select * from sysobjects where id = object_id('dbo.Articles') and sysstat &
0xf = 3)
   drop table dbo.Articles
GO

/******* del Editors before Users,Institutions */
if exists (select * from sysobjects where id = object_id('dbo.Editors') and sysstat &
0xf = 3)
   drop table dbo.Editors
GO

if exists (select * from sysobjects where id = object_id('dbo.Users') and sysstat &
0xf = 3)
   drop table dbo.Users
GO

if exists (select * from sysobjects where id = object_id('dbo.Institutions') and
sysstat & 0xf = 3)
   drop table dbo.Institutions
GO

if exists (select * from sysobjects where id = object_id('dbo.Feedback') and sysstat &
0xf = 3)
   drop table dbo.Feedback
GO

if exists (select * from sysobjects where id = object_id('dbo.VOLUMES') and sysstat &
0xf = 3)
   drop table dbo.VOLUMES
GO

if exists (select * from sysobjects where id = object_id('dbo.PrintFileTypes') and
sysstat & 0xf = 3)
   drop table dbo.PrintFileTypes
GO

/****** Object:  Table Institutions    ******/

CREATE TABLE Institutions (
   Institution_ID smallint IDENTITY(1,1) PRIMARY KEY,
   SmallDesc varchar(60) NOT NULL UNIQUE,
   AddressLine1 varchar(255),
   AddressLine2 varchar(255),
   AddressLine3 varchar(255),
   AddressLine4 varchar(255),
   AddressLine5 varchar(255),
   Telephone varchar(30),
   Facsimile varchar(30)
)
GO

/****** Object:  Table Users    ******/

CREATE TABLE Users (
   User_Id int IDENTITY(101,1) PRIMARY KEY CLUSTERED,
   EmailAddress varchar (128) NOT NULL UNIQUE,
   UserPassword varchar (32) ,
   FirstName varchar(60),
   OtherInits varchar(20),
   Surname varchar(128),
   DisplayName varchar(128),
   WebAddress varchar(128),
   UserRole smallint Default 0,
   Institution_id smallint Foreign Key references Institutions(Institution_id)
        Default 1,
   Affiliation varchar (60),
   StartDate smallDateTime DEFAULT (Getdate()),
   RenewalDate smallDateTime,
   LastAccessed smallDateTime
)
GO

IF EXISTS (SELECT name from SYSINDEXES where name = 'UserIndxEmail' )
   DROP INDEX Users.UserIndxEmail
GO

CREATE UNIQUE INDEX UserIndxEmail on Users (EMailAddress )
GO

/****** Object:  Table VOLUMES    ******/

CREATE TABLE VOLUMES (
  Volume_id smallint IDENTITY (1,1) PRIMARY KEY,
  VOLUME_NO smallint NOT NULL,
  ISSUE_NO smallint NOT NULL,
  ISSUEYear smallint NOT NULL,
  ISSUEDate smalldatetime Default (GetDate())
)
GO

/****** Object:  Table PrintFileTypes   ******/

CREATE TABLE PrintFileTypes (
   PrintType tinyint IDENTITY (1,1) PRIMARY KEY,
   PrintShortDesc varchar(20) NOT NULL UNIQUE,
   ImageFile varchar(128) Not Null,
   HtmlDescription varchar(255)
)
GO

/****** Object:  Table Editors   ******/

CREATE TABLE Editors (
  User_Id int FOREIGN KEY References Users (User_ID),
  FoundingFlag char (1)
      DEFAULT 'N'
      CONSTRAINT CK_FoundingFlag CHECK (FoundingFlag in ('Y','N')),
  ActiveFlag char (1)
      DEFAULT 'Y'
      CONSTRAINT CK_Activeflag CHECK (ActiveFlag in ('Y','N')),
)
GO

/****** Object:  Table Articles  ******/

CREATE TABLE Articles (
  Article_Id char(6) Primary Key,
  ArticleTitle varchar(255) NOT NULL,
  ReceiveDate smallDateTime Default (GetDate()),
  AcceptDate smallDateTime,
  Editor_Id int FOREIGN KEY References Users (User_ID),
  Abstract varchar (6144),
  Published char(1)
      DEFAULT 'N'
      CONSTRAINT CK_Publishedflag CHECK (Published in ('Y','N')),
  Available char(1)
      DEFAULT 'N'
```

```
      CONSTRAINT CK_Availableflag CHECK (Available in ('Y','N'))
)
GO

/****** Object:  Table ArticleAuthors  ******/

CREATE TABLE ArticleAuthors (
   Article_ID char(6) Foreign Key references Articles(Article_ID),
   User_ID int Foreign Key references Users(User_Id),
   AuthorOrder smallint Default 1,
   Institution_id smallint Foreign Key references Institutions(Institution_id)
)
GO

/****** Object:  Table ArticleKeyWords  ******/

CREATE TABLE ArticleKeyWords (
   KeyWord varchar(60)  not null,
   Article_ID char(6) Foreign Key references Articles(Article_ID)
)
Go

/****** Object:  Table ArticleMathCodes   ******/

CREATE TABLE ArticleMathCodes (
   MathCode Char(5) NOT NULL ,
   Article_ID char(6) Foreign Key references Articles(Article_ID)
)
GO

/****** Object:  Table ArticlesInFull   ******/

CREATE TABLE ArticlesInFull (
   Article_ID char(6) Foreign Key references Articles(Article_ID),
   PrintType tinyint Foreign Key references PrintFileTypes(PrintType),
   TheArticle Image
)
GO

/****** Object:  Table VolumeArticles   ******/

CREATE TABLE VolumeArticles (
   Article_ID char(6) Foreign Key references Articles(Article_ID),
   Volume_id smallint Foreign Key references Volumes(Volume_Id),
   ArticleNO tinyint
)
GO

/****** Object:  Table ArticleAbstractImages   ******/

CREATE TABLE ArticleAbstractImages (
   Article_ID char(6) Foreign Key references Articles(Article_ID),
   ImageNumber int not null,
   ImageSource varchar(255) not null,
   ImageType char(3) not null,
   ImageAltText varchar(255) not null,
   AnImage Image
)
GO

/****** Object:  Table Feedback   ******/

CREATE TABLE Feedback (
  Feedback_ID smallint IDENTITY(1,1) PRIMARY KEY,
  FeedbackDate smallDateTime DEFAULT (Getdate()),
  MessageType varchar(20) not null,
  Subject varchar(20) not null,
  SubjectOther varchar(255),
  UserName varchar(255),
  UserEmail varchar(255),
  UserTel varchar(255),
  UserFax varchar(255),
  ContactReq varchar(20) ,
  Comments varchar(1024)
)
GO



/******* Create Table Views *********/

USE JIPAM

/****** Object:  View Authors   ******/

if exists (Select table_name from Information_Schema.views
   where table_name = 'Authors')
   DROP VIEW Authors
GO

CREATE VIEW Authors
AS
Select a.Article_ID,a.AuthorOrder,a.User_ID,b.DisplayName,
      b.Surname,b.FirstName,b.OtherInits,a.Institution_id,
      b.EMailAddress,b.WebAddress
  from ArticleAuthors a JOIN USERS b on
  a.User_ID = b.User_ID
GO

/****** Object:  View ArticlesInVolume   ******/

if exists (Select table_name from Information_Schema.views
   where table_name = 'ArticlesInVolume')
   DROP VIEW ArticlesInVolume
GO

CREATE VIEW ArticlesInVolume
AS
Select Article_ID,ArticleNO,a.Volume_Id,Volume_NO,ISSUE_NO,ISSUEYEar
from VolumeArticles a INNER JOIN VOLUMES b ON
   a.Volume_ID = b.Volume_ID
GO

/****** Object:  View TitlesInVolume   ******/

if exists (Select table_name from Information_Schema.views
   where table_name = 'TitlesInVolume')
   DROP VIEW TitlesInVolume
GO

CREATE VIEW TitlesInVolume
AS
 select a.article_Id,
a.ArticleTitle,b.ArticleNO,c.AuthorOrder,c.User_Id,c.DisplayName,b.Volume_iD,b.Volume_
NO,b.Issue_NO,b.IssueYear from
    articles a INNER JOIN Authors c
 ON a.Article_ID = c.Article_Id LEFT OUTER JOIN ArticlesInVolume b
```

```
 ON a.Article_ID = b.Article_ID
 and a.Available = 'Y'
GO

/****** Object:  View TitlesUnPublished   ******/

if exists (Select table_name from Information_Schema.views
   where table_name = 'TitlesUnPublished')
   DROP VIEW TitlesUnPublished
GO

CREATE VIEW TitlesUnPublished
AS
  select a.Article_id ,a.ArticleTitle,b.User_Id,b.DisplayName,b.AuthorOrder,
    a.AcceptDate
  from articles a inner join Authors b
     on a.Article_id = b.article_id
  where a.published='N' and a.available = 'Y'
GO

/****** Object:  View EditorDetails   ******/

if exists (Select table_name from Information_Schema.views
   where table_name = 'EditorDetails')
   DROP VIEW EditorDetails
GO

CREATE VIEW EditorDetails
AS
  select a.user_id,a.FoundingFlag,
    b.FirstName,b.OtherInits,b.Surname,b.WebAddress,b.EmailAddress,b.DisplayName,
    c.AddressLine1,c.AddressLine2,c.AddressLine3,c.AddressLine4,c.AddressLine5
  from Editors a inner join users b
    on a.user_id = b.user_id
    inner join Institutions c
      on b.Institution_ID = c.Institution_ID
  where a.ActiveFlag='Y'
GO

/****** Object:  View EditorList   ******/

if exists (Select table_name from Information_Schema.views
   where table_name = 'EditorList')
   DROP VIEW EditorList
GO

CREATE VIEW EditorList
AS
  select a.user_id,b.DisplayName,b.Surname
  from Editors a inner join users b
    on a.user_id = b.user_id
GO

/****** Object:  View AuthorDetails   ******/

if exists (Select table_name from Information_Schema.views
   where table_name = 'AuthorDetails')
   DROP VIEW AuthorDetails
GO

CREATE VIEW AuthorDetails
AS
  select a.user_id,
    b.FirstName,b.OtherInits,b.Surname,b.WebAddress,b.EmailAddress,
    c.AddressLine1,c.AddressLine2,c.AddressLine3,c.AddressLine4,c.AddressLine5
  from (Select distinct user_id from ArticleAuthors) a inner join users b
    on a.user_id = b.user_id
    inner join Institutions c
      on b.Institution_ID = c.Institution_ID
GO


/***** Preload static table values   *******/


Insert into PrintFileTypes (PrintShortDesc,HtmlDescription,ImageFile) Values ('PDF
SCREEN','APPLICATION/PDF','viewpanl.jpg')
Insert into PrintFileTypes (PrintShortDesc,HtmlDescription,ImageFile) Values ('PDF
PRINTER','APPLICATION/PDF','BS00883A.gif')
GO


USE JIPAM

/******   Statistic tables Code copied from Profession ASP Techniques
            For Webmasters   *********/

/****** Object:  Table [dbo].[CountrySummary]    Script Date: 21/12/1999 14:23:29
******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[CountrySummary]')
and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[CountrySummary]
GO

/****** Object:  Table [dbo].[LangCodes]    Script Date: 21/12/1999 14:23:29 ******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[LangCodes]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[LangCodes]
GO

/****** Object:  Table [dbo].[Referers]    Script Date: 21/12/1999 14:23:29 ******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[Referers]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Referers]
GO

/****** Object:  Table [dbo].[RefererSummary]    Script Date: 21/12/1999 14:23:29
******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[RefererSummary]')
and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[RefererSummary]
GO

/****** Object:  Table [dbo].[Sessions]    Script Date: 21/12/1999 14:23:29 ******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[Sessions]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[Sessions]
GO

/****** Object:  Table [dbo].[SessionTargetSummary]    Script Date: 21/12/1999
14:23:29 ******/
if exists (select * from sysobjects where id =
object_id(N'[dbo].[SessionTargetSummary]') and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[SessionTargetSummary]
GO
```

```
/****** Object:  Table [dbo].[UserAgentSummary]    Script Date: 21/12/1999 14:23:29
******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[UserAgentSummary]')
and OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[UserAgentSummary]
GO

/****** Object:  Table [dbo].[WeekSummary]    Script Date: 21/12/1999 14:23:29 ******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[WeekSummary]') and
OBJECTPROPERTY(id, N'IsUserTable') = 1)
drop table [dbo].[WeekSummary]
GO


/****** Object:  Table [dbo].[CountrySummary]    Script Date: 21/12/1999 14:23:32
******/
CREATE TABLE [dbo].[CountrySummary] (
   [TYearNumber] [int] NOT NULL ,
   [TWeekNumber] [int] NOT NULL ,
   [ItemText] [varchar] (255) NOT NULL ,
   [ItemCount] [int] NOT NULL ,
   [HostIP] [varchar] (255) NOT NULL
) ON [PRIMARY]
GO

/****** Object:  Table [dbo].[LangCodes]    Script Date: 21/12/1999 14:23:34 ******/
CREATE TABLE [dbo].[LangCodes] (
   [LangCode] [varchar] (10) NOT NULL ,
   [LangText] [varchar] (50) NOT NULL
) ON [PRIMARY]
GO

/****** Object:  Table [dbo].[Referers]    Script Date: 21/12/1999 14:23:35 ******/
CREATE TABLE [dbo].[Referers] (
   [RefKey] [int] NOT NULL ,
   [RefURL] [varchar] (255) NULL ,
   [RefTitle] [varchar] (255) NULL ,
   [RefCount] [int] NULL ,
   [RefDelete] [int] NULL
) ON [PRIMARY]
GO

/****** Object:  Table [dbo].[RefererSummary]    Script Date: 21/12/1999 14:23:36
******/
CREATE TABLE [dbo].[RefererSummary] (
   [TYearNumber] [int] NOT NULL ,
   [TWeekNumber] [int] NOT NULL ,
   [ItemText] [varchar] (255) NOT NULL ,
   [ItemCount] [int] NOT NULL ,
   [HostIP] [varchar] (255) NOT NULL
) ON [PRIMARY]
GO

/****** Object:  Table [dbo].[Sessions]    Script Date: 21/12/1999 14:23:36 ******/
CREATE TABLE [dbo].[Sessions] (
   [EventDateTime] [datetime] NOT NULL ,
   [EventType] [varchar] (50) NULL ,
   [URL] [varchar] (255) NULL ,
   [Referer] [varchar] (255) NULL ,
   [RemoteHost] [varchar] (255) NULL ,
   [UserAgent] [varchar] (255) NULL ,
   [UALanguage] [varchar] (255) NULL ,
   [SessionCount] [int] NULL ,
   [UserID] [int] NULL ,
   [HostIP] [varchar] (255) NULL
) ON [PRIMARY]
GO

/****** Object:  Table [dbo].[SessionTargetSummary]    Script Date: 21/12/1999
14:23:37 ******/
CREATE TABLE [dbo].[SessionTargetSummary] (
   [TYearNumber] [int] NOT NULL ,
   [TWeekNumber] [int] NOT NULL ,
   [ItemText] [varchar] (255) NOT NULL ,
   [ItemCount] [int] NOT NULL ,
   [HostIP] [varchar] (255) NOT NULL
) ON [PRIMARY]
GO

/****** Object:  Table [dbo].[UserAgentSummary]    Script Date: 21/12/1999 14:23:37
******/
CREATE TABLE [dbo].[UserAgentSummary] (
   [TYearNumber] [int] NOT NULL ,
   [TWeekNumber] [int] NOT NULL ,
   [ItemText] [varchar] (255) NOT NULL ,
   [ItemCount] [int] NOT NULL ,
   [HostIP] [varchar] (255) NOT NULL
) ON [PRIMARY]
GO

/****** Object:  Table [dbo].[WeekSummary]    Script Date: 21/12/1999 14:23:38 ******/
CREATE TABLE [dbo].[WeekSummary] (
   [TYearNumber] [int] NOT NULL ,
   [TWeekNumber] [int] NOT NULL ,
   [WASessions] [int] NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[Sessions] WITH NOCHECK ADD
   CONSTRAINT [DF_Sessions_EventDateTime] DEFAULT (getdate()) FOR [EventDateTime]
GO

GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON [dbo].[CountrySummary]
TO [JipamLog]
GO

GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON [dbo].[LangCodes]  TO
[JipamLog]
GO

GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON [dbo].[Referers]  TO
[JipamLog]
GO

GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON [dbo].[RefererSummary]
TO [JipamLog]
GO

GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON [dbo].[Sessions]  TO
[JipamLog]
GO

GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON
[dbo].[SessionTargetSummary]  TO [JipamLog]
GO
```

APPENDIX B – MODULE SETUP.SQL

```
GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON [dbo].[UserAgentSummary]
TO [JipamLog]
GO

GRANT  REFERENCES ,  SELECT ,  INSERT ,  DELETE ,  UPDATE  ON [dbo].[WeekSummary]  TO
[JipamLog]
GO




/****** */
/* insert into WeekSummary (TYearNumber,TWeekNumber,WaSessions )*/
/*   Values ( DATEPART(year,GetDate()),DATEPART(week,GetDate()) - 1,0)*/
/*****/

insert into WeekSummary (TYearNumber,TWeekNumber,WaSessions )
   Values ( 2000, 25,0)

INSERT INTO langcodes (LangText, LangCode) VALUES ('Afrikaans', 'af')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (U.A.E.)', 'ar-ae')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Bahrain)', 'ar-bh')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Algeria)', 'ar-dz')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Egypt)', 'ar-eg')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Iraq)', 'ar-iq')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Jordan)', 'ar-jo')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Kuwait)', 'ar-kw')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Lebanon)', 'ar-lb')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Libya)', 'ar-ly')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Morocco)', 'ar-ma')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Oman)', 'ar-om')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Qatar)', 'ar-qa')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Saudi Arabia)', 'ar-sa')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Syria)', 'ar-sy')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Tunisia)', 'ar-tn')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Arabic (Yemen)', 'ar-ye')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Belarusian', 'be')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Bulgarian', 'bg')
GO
INSERT INTO langcodes (LangText, LangCode) VALUES ('Catalan', 'ca')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Czech', 'cs')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Danish', 'da')
INSERT INTO langcodes (LangText, LangCode) VALUES ('German (unspecified)', 'de')
INSERT INTO langcodes (LangText, LangCode) VALUES ('German (Austrian)', 'de-at')
INSERT INTO langcodes (LangText, LangCode) VALUES ('German (Swiss)', 'de-ch')
INSERT INTO langcodes (LangText, LangCode) VALUES ('German (Liechtenstein)', 'de-li')
INSERT INTO langcodes (LangText, LangCode) VALUES ('German (Luxembourg)', 'de-lu')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Greek', 'el')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (unspecified)', 'en')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (Australian)', 'en-au')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (Belize)', 'en-bz')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (Canadian)', 'en-ca')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (British)', 'en-gb')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (Ireland)', 'en-ie')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (Jamaica)', 'en-jm')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (New Zealand)', 'en-nz')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (Trinidad)', 'en-tt')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (United States)', 'en-us')
INSERT INTO langcodes (LangText, LangCode) VALUES ('English (South Africa)', 'en-za')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (unspecified)', 'es')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Argentina)', 'es-ar')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Bolivia)', 'es-bo')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Chile)', 'es-cl')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Colombia)', 'es-co')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Costa Rica)', 'es-cr')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Dominican Republic)',
'es-do')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Ecuador)', 'es-ec')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Guatemala)', 'es-gt')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Honduras)', 'es-hn')
GO
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Mexican)', 'es-mx')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Nicaragua)', 'es-ni')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Panama)', 'es-pa')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Peru)', 'es-pe')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Puerto Rico)', 'es-pr')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Paraguay)', 'es-py')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (El Salvador)', 'es-sv')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Uruguay)', 'es-uy')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Spanish (Venezuela)', 'es-ve')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Estonian', 'et')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Basque', 'eu')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Farsi', 'fa')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Finnish', 'fi')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Faeroese', 'fo')
INSERT INTO langcodes (LangText, LangCode) VALUES ('French (unspecified)', 'fr')
INSERT INTO langcodes (LangText, LangCode) VALUES ('French (Belgian)', 'fr-be')
INSERT INTO langcodes (LangText, LangCode) VALUES ('French (Canadian)', 'fr-ca')
INSERT INTO langcodes (LangText, LangCode) VALUES ('French (Swiss)', 'fr-ch')
INSERT INTO langcodes (LangText, LangCode) VALUES ('French (Luxembourg)', 'fr-lu')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Gaelic (Scottish)', 'gd')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Gaelic(Irish)', 'gd-ie')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Hebrew', 'he')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Hindi', 'hi')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Croatian', 'hr')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Hungarian', 'hu')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Indonesian', 'in')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Icelandic', 'is')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Italian (unspecified)', 'it')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Italian (Swiss)', 'it-ch')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Japanese', 'ja')
GO
INSERT INTO langcodes (LangText, LangCode) VALUES ('Yiddish', 'ji')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Korean', 'ko')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Lithuanian', 'lt')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Latvian', 'lv')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Macedonian', 'mk')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Malaysian', 'ms')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Maltese', 'mt')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Dutch (unspecified)', 'nl')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Dutch (Belgian)', 'nl-be')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Norwegian', 'no')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Polish', 'pl')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Portuguese (unspecified)', 'pt')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Portuguese (Brazilian)', 'pt-br')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Rhaeto-Romanic', 'rm')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Romanian (unspecified)', 'ro')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Romanian (Moldavia)', 'ro-mo')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Russian (unspecified)', 'ru')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Russian (Moldavia)', 'ru-mo')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Sorbian', 'sb')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Slovak', 'sk')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Slovenian', 'sl')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Albanian', 'sq')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Serbian', 'sr')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Swedish (unspecified)', 'sv')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Swedish (Finland)', 'sv-fi')
```

```
INSERT INTO langcodes (LangText, LangCode) VALUES ('Sutu', 'sx')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Sami (Lappish)', 'sz')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Thai', 'th')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Tswana', 'tn')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Turkish', 'tr')
GO
INSERT INTO langcodes (LangText, LangCode) VALUES ('Tsonga', 'ts')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Ukrainian', 'uk')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Urdu', 'ur')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Venda', 've')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Vietnamese', 'vi')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Xhosa', 'xh')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Chinese (PRC)', 'zh-cn')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Chinese (Hong Kong)', 'zh-hk')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Chinese (Singapore)', 'zh-sg')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Chinese (Taiwan)', 'zh-tw')
INSERT INTO langcodes (LangText, LangCode) VALUES ('Zulu', 'zu')
GO
```

Appendix B – Module SetUp.sql

MODULE NAME – PROCEDURE.SQL

```
USE JIPAM

/******  Internet only procedures *********/

/****** Object:  Procedure AddFeedback  ******/

if exists (select name from sysobjects where name = 'AddFeedback' and type = 'P')
   drop procedure AddFeedback
GO

CREATE PROCEDURE AddFeedback @s_MessageType varchar(20)='SUGGESTION',
    @s_Subject varchar(20) = 'OTHER',
    @s_SubjectOther varchar(255) = '',
    @s_Username varchar(255) = '',
    @s_UserEmail varchar(255) = '',
    @s_UserTel   varchar(255) = '',
    @s_UserFax   varchar(255) = '',
    @s_ContactReq varchar(20) = '',
    @s_Comments varchar(1024),
    @s_Admin varchar(255)
    AS
BEGIN
DECLARE @i_Feedbackid int
DECLARE @CMD varchar(256)
INSERT INTO Feedback (MessageType, Subject,SubjectOther,
       UserName,UserEmail,UserTel,UserFax,ContactReq,Comments)
      VALUES(@s_MessageType,
         @s_Subject,
         @s_SubjectOther,
         @s_Username,
         @s_UserEmail,
         @s_UserTel,
         @s_UserFax,
         @s_ContactReq,@s_Comments)
  IF len(@s_ContactReq) > 10 and len(@s_Admin) > 0
    BEGIN
    Select @i_Feedbackid=max(FEEDBACK_ID) from FEEDBACK
    SET @CMD = "SELECT * from FEEDBACK where FEEDBACK_ID = " & @i_Feedbackid
/****** NB need to uncomment the following line
    EXEC xp_sendmail @recipients = @s_Admin, @subject = 'JIPAM Feeback',
         @query = @CMD
*******/
    END
END
select 1
GO

/****** Object:  Procedure AddUserFull    ******/

if exists (select name from sysobjects where name = 'AddUserFull' and type = 'P')
   drop procedure AddUserFull
GO

/**** return code meanings ***
  1 Email address already exists
  >100 User_id for new entry
****/
CREATE PROCEDURE AddUserFull @s_email varchar(128),
    @s_pword varchar (32),
    @s_Firstname varchar(60) = '',
    @s_OtherInits varchar(20) = '',
    @s_Surname varchar(128) = '',
    @s_webaddress varchar(128) = '',
    @s_affiliation varchar(60) = '',
    @s_UserRole smallint = 0 AS
DECLARE @UserIdOut integer
 Select @UserIdOut=1
IF EXISTS (SELECT * FROM Users WHERE EmailAddress = @s_email)
  RETURN 1
ELSE
  BEGIN
    INSERT INTO Users (EmailAddress, UserPassword,FirstName,

OtherInits,Surname,WebAddress,Affiliation,UserRole,StartDate,RenewalDate,Lastaccessed)
      VALUES(@s_email,
         @s_pword,
         @s_Firstname,
         @s_OtherInits,
         @s_Surname,
         @s_webAddress,
         @s_affiliation,
         @s_UserRole,
         getdate(),DateAdd(mm,12,GetDate()),getdate())
  SELECT EmailAddress,FirstName,OtherInits,Surname,UserRole,User_Id
          FROM Users WHERE EmailAddress = @s_email AND UserPassword = @s_pword
  END
GO

/****** Object:  Procedure AddUserShort    ******/

if exists (select name from sysobjects where name = 'AddUserShort' and type = 'P')
   drop procedure AddUserShort
GO

/**** return code meanings ***
  1 Email address already exists
  >100 User_id for new entry
****/
CREATE PROCEDURE AddUserShort @s_user varchar(128),
    @s_pword varchar (32)As

IF EXISTS (SELECT * FROM Users WHERE EmailAddress = @s_user)
  Select 1
ELSE
  BEGIN
    INSERT INTO Users (EmailAddress, UserPassword,
       FirstName,OtherInits,Surname,StartDate,RenewalDate,Lastaccessed)
      VALUES(@s_user, @s_pword,'','','',getdate(),DateAdd(mm,12,GetDate()),Getdate())
    SELECT user_id from users where emailAddress = @s_user
  END
GO

/****** Object:  Procedure DispArtAuthors   ******/

if exists (select name from sysobjects where name = 'DispArtAuthors' and type = 'P')
   DROP PROCEDURE DispArtAuthors
GO
Create PROCEDURE DispArtAuthors
  @intArticleID char(6) = '000_00',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select b.DisplayName,b.EmailAddress,b.WebAddress,
```

```
c.smallDesc,c.AddressLine1,c.AddressLine2,c.AddressLine3,c.AddressLine4,c.AddressLine5
  from ArticleAuthors a,Users b,Institutions c
  where a.article_ID = @intArticleID
  and a.User_ID = b.User_ID
  and a.Institution_ID = c.Institution_ID
  order by c.Institution_ID
SET ROWCOUNT 0
GO

/****** Object:  Procedure DispArtDetails    ******/

if exists (select name from sysobjects where name = 'DispArtDetails' and type = 'P')
   DROP PROCEDURE DispArtDetails
GO
Create PROCEDURE DispArtDetails
  @intArticleID char(6) ='000_00',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select article_Id, ArticleTitle,ReceiveDate,AcceptDate,Editor_Id,Abstract
  from Articles where Article_ID = @intArticleID
SET ROWCOUNT 0
GO

/****** Object:  Procedure DispArtDownLoads    ******/

if exists (select name from sysobjects where name = 'DispArtDownloads' and type = 'P')
   DROP PROCEDURE DispArtDownloads
GO
Create PROCEDURE DispArtDownloads
  @intArticleID char(6) = '000_00',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select a.Article_Id,a.PrintType,b.PrintShortDesc,b.ImageFile
  from ArticlesInFull a inner join PrintFileTypes b on a.PrintType = b.PrintType
  where a.article_ID = @intArticleID
SET ROWCOUNT 0
GO

/****** Object:  Procedure DispArtKeyWords    ******/

if exists (select name from sysobjects where name = 'DispArtKeyWords' and type = 'P')
   DROP PROCEDURE DispArtKeyWords
GO
Create PROCEDURE DispArtKeyWords
  @intArticleID char(6) ='000_00',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select KeyWord
  from ArticleKeyWords
  where article_ID = @intArticleID
SET ROWCOUNT 0
GO

/****** Object:  Procedure DispArtMathCodes    ******/

if exists (select name from sysobjects where name = 'DispArtMathCodes' and type = 'P')
   DROP PROCEDURE DispArtMathCodes
GO
Create PROCEDURE DispArtMathCodes
  @intArticleID char(6)='000_00',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select MathCode
  from ArticleMathCodes
  where article_ID = @intArticleID
SET ROWCOUNT 0
GO

/****** Object:  Procedure DispAuthArt    ******/

if exists (select name from sysobjects where name = 'DispAuthArt' and type = 'P')
   DROP PROCEDURE DispAuthArt
GO
Create PROCEDURE DispAuthArt
  @intUserID int = 99,
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select article_Id, ArticleTitle,Volume_iD,Volume_NO,Issue_NO,IssueYear from
  TitlesInVolume  where User_id = @intUserID
SET ROWCOUNT 0
GO

/****** Object:  Procedure DispAuthArtList    ******/

if exists (select name from sysobjects where name = 'DispAuthArtList' and type = 'P')
   DROP PROCEDURE DispAuthArtList
GO
Create PROCEDURE DispAuthArtList
  @intUserId int = 1
AS
select c.article_id,c.user_id,c.Institution_id,c.AuthorOrder,
   d.DisplayName,
   g.ArticleTitle,
   h.Volume_id,h.Volume_No,h.Issue_no,h.IssueYear
from
(select a.article_id,a.user_id,a.Institution_id,a.AuthorOrder
   from ArticleAuthors a ,
     ( select article_id from ArticleAuthors
         where user_id = @intUserId ) b
    where a.article_id = b.article_id )c
  inner join users d on c.user_id = d.user_id
  inner join articles g on c.article_id = g.article_id
  left outer join ArticlesInVolume h on c.article_id = h.article_id
  where g.Available='Y'
order by c.article_id,c.AuthorOrder
GO

/****** Object:  Procedure DispAuthor    ******/

if exists (select name from sysobjects where name = 'DispAuthor' and type = 'P')
   DROP PROCEDURE DispAuthor
GO
Create PROCEDURE DispAuthor
  @intUserID int = 99,
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
  select a.DisplayName,a.FirstName,a.OtherInits,a.Surname,
  a.EmailAddress,a.WebAddress,

c.smallDesc,c.AddressLine1,c.AddressLine2,c.AddressLine3,c.AddressLine4,c.AddressLine5
```

```
  from Users a inner join Institutions c
     on a.Institution_id = c.Institution_id
  where a.user_id = @intUserId
SET ROWCOUNT 0
GO

/****** Object:  Procedure DispVolArticles   ******/

if exists (select name from sysobjects where name = 'DispVOLArticles' and type = 'P')
   DROP PROCEDURE DispVOLArticles
GO
Create PROCEDURE DispVOLArticles
  @intVolumeID char(6) = '000_00',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select
Article_ID,ArticleTitle,ArticleNO,User_Id,DisplayName,Volume_iD,Volume_NO,Issue_NO,Iss
ueYear
  from TitlesInVolume
  where Article_ID in (Select article_ID from VolumeArticles where Volume_Id =
@intVolumeID)
  order by ArticleNO
SET ROWCOUNT 0
GO

/****** Object:  Procedure FullEditorList   ******/

if exists (select name from sysobjects where name = 'FullEditorList' and type = 'P')
   DROP PROCEDURE FullEditorList
GO
Create PROCEDURE FullEditorList
AS
  select *
  from EditorDetails
  order by Surname
GO

/****** Object:  Procedure GetArticleFull    ******/

if exists (select name from sysobjects where name = 'GetArticleFull' and type = 'P')
   DROP PROCEDURE GetArticleFull
GO
Create PROCEDURE GetArticleFull
  @intArticleId char(6)='000_00' ,
  @intPrintType tinyint=0
AS
  select a.HtmlDescription,b.TheArticle
  from PrintFileTypes a,ArticlesInFull b
  where a.PrintType =@intPrintType and
    b.Article_ID=@intArticleId and b.PrintType =@intPrintType
GO


/****** Object:  Procedure GetArticleImage    ******/

if exists (select name from sysobjects where name = 'GetArticleImage' and type = 'P')
   drop procedure GetArticleImage
GO

CREATE PROCEDURE GetArticleImage
    @s_Article_id char(6)='000_00',
    @i_ImageNumber int = 0
As
Begin
Select ImageType,AnImage from ArticleAbstractImages where Article_Id = @s_Article_id
   and ImageNumber= @i_ImageNumber
end
go

/****** Object:  Procedure GetUserDetails   ******/
/**** Security the password must be supplied to extract the details *****/

if exists (select name from sysobjects where name = 'GetUserDetails' and type = 'P')
   DROP PROCEDURE GetUserDetails
GO
Create PROCEDURE GetUserDetails
  @intUserId int = 1,
  @strOldPass varchar(30) = ' '
AS
  select DisplayName,FirstName,OtherInits,Surname,EmailAddress,WebAddress,Affiliation
  from Users
  where user_id = @intUserId and UserPassword = @strOldPass
GO

/****** Object:  Procedure ModUserLong    ******/
/**** Security the password must be supplied to extract the details *****/

if exists (select name from sysobjects where name = 'ModUserLong' and type = 'P')
   drop procedure ModUserLong
GO
/**** return code meanings ***
  1 User_id does not exist
  0 record updated
****/
CREATE PROCEDURE ModUserLong @i_Userid int,
    @s_OldPass varchar(32),
    @s_email varchar(128),
    @s_pword varchar (32),
    @s_Firstname varchar(60) = '',
    @s_OtherInits varchar(20) = '',
    @s_Surname varchar(128) = '',
    @s_webaddress varchar(128) = '',
    @s_affiliation varchar(60) = '',
    @s_DisplayName varchar(128) = ''
    AS
BEGIN
IF NOT EXISTS (SELECT * FROM Users WHERE User_id = @i_Userid and
UserPassword=@s_OldPass)
  Select User_id  FROM Users WHERE User_id = @i_Userid and UserPassword=@s_OldPass
ELSE
  BEGIN
    Update users set EmailAddress = @s_email,
          UserPassword=@s_pword,
          FirstName=@s_Firstname,
          OtherInits=@s_OtherInits,
          Surname=@s_Surname,
          WebAddress=@s_webaddress,
          Affiliation=@s_affiliation,
          DisplayName=@s_DisplayName
        where User_id = @i_Userid
    Select User_id  FROM Users WHERE User_id = @i_Userid
  END
END
GO

/****** Object:  Procedure OrderEditorDetails   ******/
```

```
if exists (select name from sysobjects where name = 'OrderEditorDetails' and type =
'P')
   DROP PROCEDURE OrderEditorDetails
GO
Create PROCEDURE OrderEditorDetails
AS
  select user_id,DisplayName
  from EditorList
  order by displayname
GO

/****** Object:  Procedure SearchAbstracts    ******/

if exists (select name from sysobjects where name = 'SearchAbstracts' and type = 'P')
   DROP PROCEDURE SearchAbstracts
GO
Create PROCEDURE SearchAbstracts
  @strAbstract varchar(60) = 'ABCD%',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select article_Id,
ArticleTitle,User_Id,DisplayName,Volume_iD,Volume_NO,Issue_NO,IssueYear from
  TitlesInVolume
  where article_id in
  (Select Article_ID from Articles where Abstract like @strAbstract )
SET ROWCOUNT 0
GO

/****** Object:  Procedure SearchAuthors   ******/

if exists (select name from sysobjects where name = 'SearchAuthors' and type = 'P')
   DROP PROCEDURE SearchAuthors
GO
Create PROCEDURE SearchAuthors
  @strDisplayName varchar(60) = 'ABCD%',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select article_Id,
ArticleTitle,User_Id,DisplayName,Volume_iD,Volume_NO,Issue_NO,IssueYear from
  TitlesInVolume
  where User_id in
  (Select User_ID from Authors where DisplayName like @strDisplayName )
SET ROWCOUNT 0
GO

/****** Object:  Procedure SearchkeyWords    ******/

if exists (select name from sysobjects where name = 'SearchKeyWords' and type = 'P')
   DROP PROCEDURE SearchKeyWords
GO
Create PROCEDURE SearchKeyWords
  @strKeyword varchar(60) = 'ABCD%',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select article_Id,
ArticleTitle,User_Id,DisplayName,Volume_iD,Volume_NO,Issue_NO,IssueYear from
  TitlesInVolume
  where article_id in
  (Select Article_ID from ArticleKeyWords where KeyWord like @strKeyword )
SET ROWCOUNT 0
GO

/****** Object:  Procedure SearchMathCode    ******/

if exists (select name from sysobjects where name = 'SearchMathCode' and type = 'P')
   DROP PROCEDURE SearchMathCode
GO
Create PROCEDURE SearchMathCode
  @strMathCode varchar(5) = 'ABCDE',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select article_Id,
ArticleTitle,User_Id,DisplayName,Volume_iD,Volume_NO,Issue_NO,IssueYear from
  TitlesInVolume
  where article_id in
  (Select Article_ID from ArticleMathCodes where MathCode like @strMathCode )
SET ROWCOUNT 0
GO

/****** Object:  Procedure SearchTitles    ******/

if exists (select name from sysobjects where name = 'SearchTitles' and type = 'P')
   DROP PROCEDURE SearchTitles
GO
Create PROCEDURE SearchTitles
  @strTitleName varchar(60) = 'ABCD%',
  @maxrowCount int = 100
AS
SET ROWCOUNT @maxrowCount
select article_Id,
ArticleTitle,User_Id,DisplayName,Volume_iD,Volume_NO,Issue_NO,IssueYear from
  TitlesInVolume
   where articletitle like @strTitleName
SET ROWCOUNT 0
GO


/****** Object:  Procedure SendUserPassword    ******/

if exists (select name from sysobjects where name = 'SendUserPassword' and type = 'P')
   drop procedure SendUserPassword
GO
CREATE PROCEDURE SendUserPassword
    @s_user varchar(128) as
IF NOT EXISTS (SELECT * FROM Users WHERE EmailAddress = @s_user)
  Select 1
ELSE
  BEGIN
    DECLARE @strPass varchar(32)
    select @strPass=Userpassword from users where emailaddress = @s_user
    EXEC xp_sendmail @recipients = @s_user, @subject = 'JIPAM Password', @message =
@strPass
    select 0
  END
go

/****** Object:  Procedure UnPubArticles    ******/

if exists (select name from sysobjects where name = 'UnPubArticles' and type = 'P')
   DROP PROCEDURE UnPubArticles
GO
Create PROCEDURE UnPubArticles
```

```
AS
  select article_id,ArticleTitle,User_Id,DisplayName,AuthorOrder,
    AcceptDate
  from TitlesUnPublished
  order by AcceptDate,AuthorOrder
GO

/****** Object:  Procedure ValidateUser    ******/

if exists (select name from sysobjects where name = 'ValidateUser' and type = 'P')
   drop procedure ValidateUser
GO

CREATE PROCEDURE ValidateUser
    @s_user varchar(128), @s_pword varchar (32) AS
BEGIN
 IF EXISTS (SELECT * FROM Users WHERE EmailAddress = @s_user AND UserPassword =
@s_pword)
      BEGIN
      Declare @RenDate smallDateTime
      Select @RenDate = RenewalDate from Users where  EmailAddress = @s_user
      if @RenDate < GetDate()
         begin
            Update Users Set RenewalDate = DateAdd(mm,12,@RenDate) WHERE EmailAddress
= @s_user
         end
      UPdate USers Set LastAccessed = GetDate() WHERE EmailAddress = @s_user
      END
SELECT EmailAddress,FirstName,OtherInits,Surname,UserRole,User_Id FROM Users WHERE
EmailAddress = @s_user AND UserPassword = @s_pword
END
GO

/****** Object:  Procedure VolumeList    ******/

if exists (select name from sysobjects where name = 'VOLumeList' and type = 'P')
   DROP PROCEDURE VOLumeList
GO
Create PROCEDURE VOLumeList
  @intVolumeID int = 1
AS
select Volume_id,VOLUME_NO,ISSUE_NO,ISSUEYear
  from VOLUMES
  where Volume_id <> @intVolumeID
GO


Grant Execute On AddFeedback to internet
Grant Execute On AddUserFull  TO internet
Grant Execute On AddUserShort  TO internet
Grant Execute On DispArtAuthors to internet
Grant Execute On DispArtDetails to internet
Grant Execute On DispArtDownloads to internet
Grant Execute On DispArtKeyWords to internet
Grant Execute On DispArtMathCodes to internet
Grant Execute On DispAuthArt to internet
Grant Execute On DispAuthor to internet
Grant Execute On DispAuthArtList to internet
Grant Execute On DispVOLArticles to internet
Grant Execute On FullEditorList to internet
Grant Execute On GetArticleFull to internet
Grant Execute On GetArticleImage to internet
Grant Execute On GetUserDetails to internet
Grant Execute On ModUserLong  TO internet
Grant Execute On OrderEditorDetails to internet
Grant Execute On SearchAbstracts to internet
Grant Execute On SearchAuthors to internet
Grant Execute On SearchKeyWords to internet
Grant Execute On SearchMathCode to internet
Grant Execute On SearchTitles to internet
Grant Execute On SendUserPassword to internet
Grant Execute On UnPubArticles to internet
Grant Execute On ValidateUser  TO  internet
Grant Execute On VOLumeList to internet
go

/******* Admin Only procedures **********/

/****** Object:  Procedure AddUserFullAdmin    ******/

if exists (select name from sysobjects where name = 'AddUserFullAdmin' and type = 'P')
   drop procedure AddUserFullAdmin
GO

/**** return code meanings ***
  1 Email address already exists
  >100 User_id for new entry
****/
CREATE PROCEDURE AddUserFullAdmin
    @s_email varchar(128),
    @s_pword varchar (32),
    @s_Firstname varchar(60) = '',
    @s_OtherInits varchar(20) = '',
    @s_Surname varchar(128) = '',
    @s_webaddress varchar(128) = '',
    @s_affiliation varchar(60) = '',
    @s_UserRole smallint = 0,
    @s_InstitutionID smallint =1  AS
DECLARE @UserIdOut integer
 Select @UserIdOut=1

IF EXISTS (SELECT * FROM Users WHERE EmailAddress = @s_email)
  RETURN 1
ELSE
  BEGIN
    INSERT INTO Users (EmailAddress, UserPassword,FirstName,
       OtherInits,Surname,WebAddress,Affiliation,UserRole,Institution_ID,
       StartDate,RenewalDate,Lastaccessed)
      VALUES(@s_email,
         @s_pword,
         @s_Firstname,
         @s_OtherInits,
         @s_Surname,
         @s_webAddress,
         @s_affiliation,
         @s_UserRole,
         @s_InstitutionID,
         getdate(),DateAdd(mm,12,GetDate()),getdate())
    SELECT  @UserIdOut=user_id from users where emailAddress = @s_email
    RETURN @UserIdOut
  END
GO

/****** Object:  Procedure ModArticle    ******/

if exists (select name from sysobjects where name = 'ModArticle' and type = 'P')
   drop procedure ModArticle
```

```
GO

CREATE PROCEDURE ModArticle
    @s_Article_id char(6)='000_00',
    @s_ArticleTitle varchar(255),
    @s_ReceiveDate varchar(32),
    @s_AcceptDate varchar(32),
    @i_Editor_Id int,
    @s_Abstract varchar(6144),
    @s_Published char(1),
    @s_Available char(1)
    AS
SET DATEFORMAT dmy
IF NOT EXISTS (SELECT Article_Id FROM Articles WHERE Article_Id = @s_Article_id)
  Return 1
ELSE
  BEGIN
    Update Articles set
          ArticleTitle = @s_ArticleTitle,
          ReceiveDate=@s_ReceiveDate,
          AcceptDate=@s_AcceptDate,
          Editor_Id=@i_Editor_Id,
          Abstract=@s_Abstract,
          Published=@s_Published,
          Available=@s_Available
        where Article_id = @s_Article_id
    Return 0
  END
GO


/****** Object:  Procedure AddArticle   ******/

if exists (select name from sysobjects where name = 'AddArticle' and type = 'P')
   drop procedure AddArticle
GO

CREATE PROCEDURE AddArticle
    @s_Article_id char(6)='000_00',
    @s_ArticleTitle varchar(255),
    @s_ReceiveDate varchar(32),
    @s_AcceptDate varchar(32),
    @i_Editor_Id int,
    @s_Abstract varchar(6144),
    @s_Published char(1),
    @s_Available char(1)
    AS

SET DATEFORMAT dmy
BEGIN
INSERT INTO Articles
(Article_id,ArticleTitle,ReceiveDate,AcceptDate,Editor_Id,Abstract,Published,Available
)
    VALUES
(@s_Article_id,@s_ArticleTitle,@s_ReceiveDate,@s_AcceptDate,@i_Editor_Id,@s_Abstract,@
s_Published,@s_Available)
END
GO

/****** Object:  Procedure GetNextArticleId    ******/

if exists (select name from sysobjects where name = 'GetNextArticleId' and type = 'P')
   drop procedure GetNextArticleId
GO
```

```
CREATE PROCEDURE GetNextArticleId
  @sghtyear char(6) output
  as
BEGIN
  declare @article_id int
  declare @year char(4)
  Set @Year = cast(datepart(year,getdate()) as char(4))
  Set @sghtyear = '____' + substring(@year,3,2)
  select @article_id=max(cast(substring(article_id,1,3) as int)) from articles where
article_id like @sghtyear
  set @article_id = @article_id+1001
  Set @sghtyear = substring(cast(@article_id as char(4)),2,3) + '_' +
substring(@year,3,2)

END
GO

/****** Object:  Procedure AddVolume   ******/

if exists (select name from sysobjects where name = 'AddVolume' and type = 'P')
   drop procedure AddVolume
GO

CREATE PROCEDURE AddVolume
    @i_VolumeNo smallint,
    @i_ISSUE_NO smallint,
    @i_ISSUEYear smallint,
    @d_ISSUEDate varchar(20)
    AS
DECLARE @intVolumeId integer

SET DATEFORMAT dmy
BEGIN
IF EXISTS (SELECT Volume_id FROM VOLUMES WHERE Volume_No = @i_VolumeNo
     and ISSUE_NO = @i_ISSUE_NO and ISSUEYear = @i_ISSUEYear)
  Return 0
ELSE
BEGIN
 INSERT INTO VOLUMES (Volume_No,ISSUE_NO,ISSUEYear,ISSUEDate)
     VALUES (@i_VolumeNo,@i_ISSUE_NO,@i_ISSUEYear,@d_ISSUEDate)
 SELECT @intVolumeId=Volume_id FROM VOLUMES WHERE Volume_No = @i_VolumeNo  and
ISSUE_NO = @i_ISSUE_NO and ISSUEYear = @i_ISSUEYear
  Return @intVolumeId
END
END
GO

/****** Object:  Procedure ModVolume    ******/

if exists (select name from sysobjects where name = 'ModVolume' and type = 'P')
   drop procedure ModVolume
GO

CREATE PROCEDURE ModVolume
    @i_VolumeNo smallint,
    @i_ISSUE_NO smallint,
    @i_ISSUEYear smallint,
    @d_ISSUEDate varchar(20),
    @i_VolumeId  smallint
    AS
SET DATEFORMAT dmy
BEGIN
```

```
IF Not EXISTS (SELECT Volume_id FROM VOLUMES WHERE Volume_id = @i_VolumeId )
  Return 0
ELSE
BEGIN
    Update VOLUMES set
          Volume_No = @i_VolumeNo,
          ISSUE_NO=@i_ISSUE_NO,
          ISSUEYear=@i_ISSUEYear,
          ISSUEDate=@d_ISSUEDate
        where Volume_id = @i_VolumeId
  Return @i_VolumeId
END
END
GO

/****** Object:  Procedure AddInstitution    ******/

if exists (select name from sysobjects where name = 'AddInstitution' and type = 'P')
   drop procedure AddInstitution
GO

CREATE PROCEDURE AddInstitution
   @s_SmallDesc varchar(60),
   @s_AddressLine1 varchar(255),
   @s_AddressLine2 varchar(255),
   @s_AddressLine3 varchar(255),
   @s_AddressLine4 varchar(255),
   @s_AddressLine5 varchar(255),
   @s_Telephone varchar(30),
   @s_Facsimile varchar(30)
    AS
DECLARE @intInstitutionId integer

SET DATEFORMAT dmy
BEGIN
IF EXISTS (SELECT Institution_ID FROM Institutions  WHERE SmallDesc = @s_SmallDesc)
  Return 0
ELSE
BEGIN
 INSERT INTO Institutions
(SmallDesc,AddressLine1,AddressLine2,AddressLine3,AddressLine4,
      AddressLine5,Telephone,Facsimile)
     VALUES (@s_SmallDesc,@s_AddressLine1,@s_AddressLine2,@s_AddressLine3,
         @s_AddressLine4,@s_AddressLine5,@s_Telephone,@s_Facsimile)
 SELECT @intInstitutionId=Institution_ID FROM Institutions  WHERE SmallDesc =
@s_SmallDesc
  Return @intInstitutionId
END
END
GO

/****** Object:  Procedure ModInstitution   ******/

if exists (select name from sysobjects where name = 'ModInstitution' and type = 'P')
   drop procedure ModInstitution
GO

CREATE PROCEDURE ModInstitution
   @s_SmallDesc varchar(60),
   @s_AddressLine1 varchar(255),
   @s_AddressLine2 varchar(255),
   @s_AddressLine3 varchar(255),
   @s_AddressLine4 varchar(255),
   @s_AddressLine5 varchar(255),
   @s_Telephone varchar(30),
   @s_Facsimile varchar(30),
   @i_Institution_ID smallint
    AS
DECLARE @intInstitutionId integer

SET DATEFORMAT dmy
BEGIN
SELECT @intInstitutionId=Institution_ID FROM Institutions  WHERE SmallDesc =
@s_SmallDesc
if @intInstitutionId <> @i_Institution_ID
  Return -1

IF Not EXISTS (SELECT Institution_ID FROM Institutions  WHERE Institution_ID =
@i_Institution_ID)
  Return 0

UPdate Institutions Set
    SmallDesc=@s_SmallDesc,
    AddressLine1=@s_AddressLine1,
    AddressLine2=@s_AddressLine2,
    AddressLine3=@s_AddressLine3,
    AddressLine4=@s_AddressLine4,
    AddressLine5=@s_AddressLine5,
    Telephone=@s_Telephone,
    Facsimile=@s_Facsimile
    WHERE Institution_ID = @i_Institution_ID
  Return @i_Institution_ID
END
GO



/***** The procedures for the statistic tables(DeleteOldRecords and SummariseSession
          are taken from the book Professional ASP Techniques for Webmasters  ******/

/****** Object:  Stored Procedure dbo.DeleteOldRecords  ******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[DeleteOldRecords ]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[DeleteOldRecords]
GO

/****** Object:  Stored Procedure dbo.SummariseSession******/
if exists (select * from sysobjects where id = object_id(N'[dbo].[SummariseSession]')
and OBJECTPROPERTY(id, N'IsProcedure') = 1)
drop procedure [dbo].[SummariseSession]
GO


SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

/****** Object:  Stored Procedure dbo.DeleteOldRecords  ******/

CREATE PROCEDURE DeleteOldRecords AS
DECLARE @rowsupdated int
BEGIN TRANSACTION
DELETE
FROM CountrySummary
WHERE (TYearNumber = DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) - 25))
   OR (TYearNumber < DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) + 26))
```

```
IF (@@ERROR <> 0) GOTO on_error

DELETE
FROM RefererSummary
WHERE (TYearNumber = DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) - 25))
   OR (TYearNumber < DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) + 26))
IF (@@ERROR <> 0) GOTO on_error

DELETE
FROM SessionTargetSummary
WHERE (TYearNumber = DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) - 25))
   OR (TYearNumber < DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) + 26))
IF (@@ERROR <> 0) GOTO on_error

DELETE
FROM UserAgentSummary
WHERE (TYearNumber = DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) - 25))
   OR (TYearNumber < DATEPART(year, GETDATE()) AND TWeekNumber < (DATEPART(week,
GETDATE()) + 26))
IF (@@ERROR <> 0) GOTO on_error

SELECT 'Complete Deletion'
COMMIT TRANSACTION
RETURN (0)
on_error:
SELECT 'Error - deletion aborted.'
ROLLBACK TRANSACTION
RETURN(1)


GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

GRANT  EXECUTE  ON [dbo].[DeleteOldRecords]  TO [JipamLog]
GO

SET QUOTED_IDENTIFIER  ON    SET ANSI_NULLS  ON
GO

/****** Object:  Stored Procedure dbo.SummariseSession    Script Date: 27/04/98
09:22:37 ******/
CREATE proc SummariseSession as
DECLARE @ToYear int
DECLARE @ToWeek int
DECLARE @ThisYear int
DECLARE @ThisWeek int
DECLARE @WASessions int

BEGIN TRANSACTION

SELECT @ToYear = DATEPART(year, GETDATE()),
       @ToWeek = DATEPART(week, GETDATE())

SELECT @ThisYear = MAX(TYearNumber)
FROM WeekSummary

SELECT @ThisWeek = MAX(TWeekNumber)
FROM WeekSummary
WHERE TYearNumber = @ThisYear

SELECT @ThisWeek = @ThisWeek + 1
IF (@ThisWeek > 53)
  BEGIN
    SELECT @ThisWeek = 1
    SELECT @ThisYear = @ThisYear + 1
  END

WHILE (@ThisYear = @ToYear AND @ThisWeek < @ToWeek) OR (@ThisYear < @ToYear)
BEGIN

SELECT @WASessions = COUNT(EventDateTime)
FROM Sessions
WHERE DATEPART(year, EventDateTime) = @ThisYear
      AND DATEPART(week, EventDateTime) = @ThisWeek
      AND EventType = 'New Session'

IF (@@ERROR <> 0) GOTO on_error

INSERT INTO WeekSummary
SELECT @ThisYear,
       @ThisWeek,
       ISNULL(@WASessions, 0)

IF (@@ERROR <> 0) GOTO on_error

SELECT @ThisWeek = @ThisWeek + 1
IF (@ThisWeek > 53)
  BEGIN
    SELECT @ThisWeek = 1
    SELECT @ThisYear = @ThisYear + 1
  END
/***** start loop over again *****/
END

INSERT INTO RefererSummary
SELECT RefYear = MAX(DATEPART(year, EventDateTime)),
       RefWeek = MAX(DATEPART(week, EventDateTime)),
       RefURL=MAX(
         CASE WHEN CHARINDEX('?', Referer) > 5
         THEN SUBSTRING(Referer, 1, CHARINDEX('?', Referer) - 1)
         ELSE Referer
         END),
       RefCount=COUNT(Referer),
       RefHostIP = MAX(HostIP)
FROM Sessions
WHERE DATEDIFF(week, EventDateTime, GETDATE()) > 0
      AND Referer IS NOT NULL       AND Referer <> ''
      AND CHARINDEX('Jipam/', Referer) = 0
      AND CHARINDEX('file:', Referer) = 0
GROUP BY CASE WHEN CHARINDEX('?', Referer) > 5
         THEN SUBSTRING(Referer, 1, CHARINDEX('?', Referer) - 1)
         ELSE Referer
         END,
         DATEPART(year, EventDateTime),
         DATEPART(week, EventDateTime),
         HostIP

IF (@@ERROR <> 0) GOTO on_error

INSERT INTO SessionTargetSummary
```

APPENDIX B – MODULE PROCEDURE.SQL

```
SELECT TargYear = MAX(DATEPART(year, EventDateTime)),
       TargWeek = MAX(DATEPART(week, EventDateTime)),
       TargText = MAX(URL),
       TargCount = COUNT(URL),
       TargHostIP = MAX(HostIP)
FROM Sessions
WHERE DATEDIFF(week, EventDateTime, GETDATE()) > 0
      AND URL IS NOT NULL
      AND URL <> ''
GROUP BY URL,
         DATEPART(year, EventDateTime),
         DATEPART(week, EventDateTime),
         HostIP

IF (@@ERROR <> 0) GOTO on_error

INSERT INTO UserAgentSummary
SELECT UAYear = MAX(DATEPART(year, EventDateTime)),
       UAWeek = MAX(DATEPART(week, EventDateTime)),
       UAText = MAX(UserAgent),
       UACount = COUNT(UserAgent),
       UAHostIP = MAX(HostIP)
FROM Sessions
WHERE DATEDIFF(week, EventDateTime, GETDATE()) > 0
      AND UserAgent IS NOT NULL
      AND UserAgent <> ''
GROUP BY UserAgent,
         DATEPART(year, EventDateTime),
         DATEPART(week, EventDateTime),
         HostIP

IF (@@ERROR <> 0) GOTO on_error

INSERT INTO CountrySummary
SELECT CoYear = MAX(DATEPART(year, EventDateTime)),
       CoWeek = MAX(DATEPART(week, EventDateTime)),
       CoText = MAX(
         CASE WHEN CHARINDEX(',', UALanguage) > 1
         THEN SUBSTRING(UALanguage, 1, CHARINDEX(',', UALanguage) - 1)
         ELSE UALanguage
         END),
       CoCount = COUNT(UALanguage),
       CoHostIP = MAX(HostIP)
FROM Sessions
WHERE DATEDIFF(week, EventDateTime, GETDATE()) > 0
      AND UALanguage IS NOT NULL
      AND UALanguage <> ''
GROUP BY CASE WHEN CHARINDEX(',', UALanguage) > 1
         THEN SUBSTRING(UALanguage, 1, CHARINDEX(',', UALanguage) - 1)
         ELSE UALanguage
         END,
         DATEPART(year, EventDateTime),
         DATEPART(week, EventDateTime),
         HostIP

IF (@@ERROR <> 0) GOTO on_error

DELETE
FROM Sessions
WHERE DATEDIFF(week, EventDateTime, GETDATE()) > 0

IF (@@ERROR <> 0) GOTO on_error

COMMIT TRANSACTION
SELECT 'Sessions summary succeeded.'

RETURN (0)

on_error:
ROLLBACK TRANSACTION
SELECT 'Error - summary aborted.'RETURN (1)

GO

SET QUOTED_IDENTIFIER  OFF    SET ANSI_NULLS  ON
GO

GRANT  EXECUTE  ON [dbo].[SummariseSession]  TO [JipamLog]
GO


/***** Create a Job to automatically run statistic table summarisation and clean up
******/

USE MSDB
/*****  must be in this database *******/

if exists (Select * from sysjobs where name = 'Weekly JIPAM Session Update2')
begin
   Exec sp_delete_Job @job_name = 'Weekly JIPAM Session Update2'
end

EXEC sp_add_job @job_name = 'Weekly JIPAM Session Update2',
    @enabled = 1,
    @description = 'weekly update of summary tables'

EXEC sp_add_jobstep @job_name = 'Weekly JIPAM Session Update2',
    @step_name = 'Delete records older than 26 weeks',
    @subsystem = 'TSQL',
    @database_name ='JIPAM',
    @command = 'exec DeleteOldRecords',
    @on_success_action = 3,
    @retry_attempts = 5,
    @retry_interval = 5

EXEC sp_add_jobstep @job_name = 'Weekly JIPAM Session Update2',
    @step_name = 'Weekly Update of Summary Tables',
    @subsystem = 'TSQL',
    @database_name ='JIPAM',
    @command = 'exec SummariseSession',
    @retry_attempts = 5,
    @retry_interval = 5

EXEC sp_add_jobschedule @job_name = 'Weekly JIPAM Session Update2',
    @name = 'UpdateSession',
    @freq_type = 4,
    @freq_interval = 1,
    @active_start_time = 130000
```

Appendix B – Module Procedure.sql

APPENDIX C

JIPAM WEB SITE
Dynamic Database System

Web Server ASP Code

MODULE NAME – ACCEPTED.ASP

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Accepted</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="ACCEPTED"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<Hl ALIGN="center"><FONT
COLOR="#FF0000">N</FONT>ew <FONT COLOR="#FF0000">P</FONT>apers --
<FONT COLOR="#FF0000">P</FONT>apers <FONT COLOR="#FF0000">A</FONT>ccepted
<FONT COLOR="#FF0000">F</FONT>or <FONT COLOR="#FF0000">P</FONT>ublication
</H1>
<P ALIGN="center"> </P>
<P ALIGN="left">Below we list the papers that have been accepted for
publication in JIPAM but have not yet made it to to an issue. </P>
<P ALIGN="left"> The list is periodically updated and will change as papers
become accepted or issues are published. Some titles link to a page that
describes the paper in more detail. On that page, you can also register to be
notified when the paper is published. </P>
<P ALIGN="left">Be aware that some papers may not be in final form until
publication.  In order to conform with the journal requirements, titles,
abstracts, key words and AMS subject classification numbers may change
slightly.</P>
<%
   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   oCon.Open strConnection

   'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = "UnPubArticles"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute( lngRecsAffected)
   if oRs.BOF then 'this could happen but is unlikely
%>
<P>There are no papers accepted that are unpublished.</P><P></P><HR>
<!-- #include virtual="/jipam/common/Footer.inc" -->

<%
     Response.End
   end if

%>
<P ALIGN="left">The list below is sorted by date of acceptance.</P>
<DIV ALIGN="center">
<CENTER>
<TABLE BORDER="2" CELLPADDING="2" WIDTH="85%" CELLSPACING="14">
<%
    intArticleID=-1
    strAuthorList=""
    do while not oRs.Eof
      strAuthor = oRs.Fields("DisPlayName")
      intAuthorId = oRs.Fields("User_Id")
      intArticleId1 = oRs.Fields("article_id")

      if intArticleID = intArticleId1 then 'just another author
        strAuthorList = strAuthorList & ", <A HREF=DISPAUTHTITLE.ASP?UI=" &
intAuthorId & ">" & strAuthor & "</A>"

      else ' new article
         if len(strAuthorList) then ' complete the previous article

%>
<P ALIGN="center"><%=strAuthorList%></P>
</TD>
</TR>
<%
         end if
        strArticleTitle = oRs.Fields("ArticleTitle")
        intArticleID = oRs.Fields("Article_ID")
        strDate= FormatDateTime(oRs.Fields("AcceptDate"),vbShortDate)
%>
<TR>
<TD WIDTH="14%" VALIGN="top" ALIGN="center" BGCOLOR="#FFFFCC"><%=strDate%>.</TD>
<TD WIDTH="86%" ALIGN="center" BGCOLOR="#FFFFCC"> <P ALIGN="center">
<A HREF="DISPARTICLE.ASP?AI=<%=intArticleId%>"><B><I><FONT COLOR="#000080">
<%=strArticleTitle%></FONT></I></B></A></P>
<%
        strAuthorList = "<A HREF=DISPAUTHTITLE.ASP?UI=" & intAuthorId & ">" &
strAuthor & "</A>"
      end if ' of new article
      oRs.MoveNext
    loop
    Set oRs=nothing
    Set oCmd=nothing
    ' complete author list
%>
<P ALIGN="center"><%=strAuthorList%></P>
</TD>
</TR>
</TABLE>
</CENTER>
</DIV>
<HR>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

APPENDIX C – MODULE ACCEPTED.ASP

Module Name – AimsScope.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Aims and Scope</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="SCOPE"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<H1 ALIGN="center">
<FONT COLOR="#FF0000">A</FONT>ims and <FONT COLOR="#FF0000">S</FONT>cope
</H1>
<div align="left">
  <table border="0" cellpadding="17" cellspacing="0" width="100%">
    <tr>
      <td width="100%">
        <h1 align="center"><font color="#FF0000">A</font>ims and <font
color="#FF0000">S</font>cope</h1>
        <p align="left">The peer-reviewed electronic Journal of Inequalities in Pure
and
        Applied Mathematics (JIPAM) aims to foster and develop further growth in
        all areas of mathematics relating to inequalities. The journal accepts
        high quality papers containing original research results, survey
        articles of exceptional merit, short letters and notes. The journal
        recognizes the need for papers to be published in a timely fashion and
        therefore papers will be refereed within <b>100 working days of
        submission</b>. Resubmitted papers will be refereed within 45 working
        days of submission. The journal encourages <a
href="authors.asp#Submissions">submissions</a>
        in electronic form, typed form or clearly handwritten English. To
        encourage clear and succinct papers, the journal will award, annually, a
        <blink><a href="authors.asp#Prize">prize</a></blink> for the
        best-published paper. The <a href="http://rgmia.vu.edu.au/SSDragomirWeb.html"
target="_self">Editor-in-Chief</a>
        together with a panel of international experts will judge the published
        papers.</p>
        <p align="left">JIPAM will initially be published twice a year in March
        and October.</td>
    </tr>
  </table>
</div>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module AimsScope.asp

Module Name – Authors.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Authors</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="AUTHORS"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<table border="0" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td width="100%">
      <h1 align="center"><font color="#FF0000">I</font>nformation
       for  <font color="#FF0000">A</font>uthors</h1>
      <ul>
        <li><a href="#Submissions">Submissions</a></li>
        <li><a href="#Preparation">Article Preparation</a></li>
        <li><a href="copyright.asp">Copyright</a></li>
        <li><a href="#Prize">Prize</a></li>
        <li><a href="#Other">Other Information</a></li>
      </ul>
      <table border="0" cellpadding="0" cellspacing="0" width="100%">
        <tr>
          <td width="5%"></td>
          <td width="90%">
            <h1 align="center"><a name="Submissions"></a><font
color="#FF0000">S</font>ubmissions</h1>
            <p align="left">It is understood that when submitting a manuscript
            the author confirms that it has not been previously published or is
            being considered for publication elsewhere. Please visit the <a
href="copyright.asp"> copyright page</a> for more details. </p>
            <p align="left">Manuscripts may be submitted to any of the <a
href="editors.asp">editors</a>.
            In addition, <b>you must send a copy of your manuscript to the <a
href="contactus.asp">editorial
            office</a></b> along with a covering note telling us which editor
            your manuscript is with. You may of course send the manuscript
            directly to the <a href="contactus.asp">editorial office</a> and the
            Editor-in-Chief can decide on the appropriate editor.</p>
            <p align="left">The <a href="contactus.asp">editorial office</a>
            prefers electronic submission, but we will accept three typed
            copies. The intending author should check with their preferred <a
href="editors.asp">editor</a>
            regarding submission to them.</p>
            <p align="left">The publisher strongly encourages the electronic
            submission of manuscripts written in LaTeX or AMSLaTeX.  An
            accompanying DVI file should also be included.</p>
            <p align="left"><b>Contributions will be reviewed in 100 working
            days</b>.  If a manuscript is re-submitted, the author will be
            notified within 45 working days.  Upon acceptance,
            publication will usually occur within two issues.</p>
            <p align="left">The author will be provided with a proof for correction
after
            acceptance.</p>
            <p align="left"> </p>
            <h1 align="center"><font color="#FF0000"><a
name="Preparation"></a>A</font>rticle
            <font color="#FF0000">P</font>reparation</h1>
            <p align="left">Manuscripts must be written in English.</p>
            <p align="left">The front-matter should include the title of the
            article, author's name, affiliation, e-mail, web page address and
            postal address. A running head of less than sixty characters should
            also be given.  You should also include Mathematics Subject
            Classification codes <a href="http://www.ams.org/mathweb/msc2000.html"
target="_top">(MSC2000)</a>
            as well as a  list of keywords.</p>
            <p align="left">The manuscript must be accompanied by a brief
            abstract no longer than two hundred words. The use of mathematical
            symbols and displayed formulae must be kept to a
            minimum.   References to theorems, equations etc. must not
            appear in the abstract.  Bibliographic references, if vital,
            should be written in full.</p>
            <p align="left">All figures and tables must be centred and labeled. 
The position of figures should be clearly indicated
            in the manuscript and attached in a form suitable for direct
            reproduction.   Preference is given to figures produced in
            EPS or PDF format.  Bitmap formats may be acceptable (in this
            case we would prefer JPG or PNG)  if their resolution does not
            preclude high quality print reproduction.</p>
            <p align="left">All equations, theorems, tables, figures should be
            referenced as in &quot;Refer to Figure 3.2&quot;, or &quot;.. this
            concludes the proof of Theorem  3.1&quot;.  Bibliographic
            references should be numbered and listed alphabetically at the end
            of the article. Authors are advised to consult Mathematical Reviews
            for the standard abbreviations of journal names. </p>
            <p align="left">Click <a href="jipam.zip">here</a> to download a
            sample article.  You will need AMSLaTeX and the LaTeX graphics
            package to compile the TeX source code.  DVI, postscript (PS)
            and PDF versions of the paper are also included.  Refer to this
            sample for examples of our required formats.</p>
            <p align="left">Submission should also be accompanied by a signed <a
href="common/copyright.pdf">"Transfer
            of Copyright" form</a>. The transfer of copyright will come into affect
upon
            acceptance of the manuscript. Visit the <a href="copyright.asp"> copyright
page</a> for
            more details.</p>
            <p align="left"> </p>
            <h1 align="center"><a name="Prize"></a><font
color="#FF0000">P</font>rize</h1>
            <p align="left">JIPAM would like to encourage the submission of
            manuscripts of the highest quality. To this end, JIPAM will award an
            annual prize to the best published paper. The <a
href="editors.asp">Editor-in-Chief</a>
            together with a panel of international experts will judge the
            papers.  The winner will be announced on the following issue of
            the journal, on the <a href="http://rgmia.vu.edu.au">RGMIA</a> web
            page and the RGMIA mailing list.</p>
            <p align="left">Further details will be provided in due course.</p>
            <p align="left"> </p>
            <h1 align="center"><a name="Other"></a><font color="#FF0000">O</font>ther
            <font color="#FF0000">I</font>nformation</h1>
            <p align="left">Authors of published papers without e-mail
            facilities will be sent thirty off-prints of their published
            article. Authors will also have the right to use the electronic
            version of the JIPAM article on one personal web page.</p>
            <p align="left">For further information contact the <a
href="contactUs.asp">Managing
            Editor</a>.</td>
```

Appendix C – Module Authors.asp

```
          <td width="5%"></td>
        </tr>
      </table>
    </td>
  </tr>
</table>


<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module Authors.asp

MODULE NAME – COMMON.INC

```
<%
  Dim strConnection
  Dim strConnLogs
  Dim MaxLines
  Dim MathCodeUrl
  Dim LatestVolumId
  LatestVolumeId=Application.Contents.Item("LatestVolumId")
  strConnection = Application.Contents.Item("strConnection")
  strConnLogs = Application.Contents.Item("strConnLogs")
  MaxLines=Application.Contents.Item("MaxLines")
  MathCodeUrl= Application.Contents.Item("MathCodeUrl")

Function IsAdmin()
  IsAdmin = false
  if Session("UseriD")>0 and Session("UserRole") >= 4 then IsAdmin=true
end Function
Function IsLoggedOn()
  IsLoggedOn=false
  if Session("UseriD")>0 then IsLoggedOn=true
end Function

%>
```

Module Name – ContactUs.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Contact Us</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="CONTACT"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<H1 ALIGN="center">
<FONT COLOR="#FF0000">C</FONT>ontact <FONT COLOR="#FF0000">U</FONT>s
</H1>
<table border="0" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td width="100%">
      <h1 align="center"><font color="#FF0000">C</font>ontact <font
color="#FF0000">D</font>etails</h1>
      <p align="center"> </p>
      <blockquote>
        <p><strong>Editor-in-Chief:</strong></p>
      </blockquote>
      <table border="0" cellpadding="0" cellspacing="0" width="100%"
bgcolor="#FFFFCC">
        <tr>
          <td width="20%"> </td>
          <td width="80%"><font color="#C60000"><strong>Sever S.
Dragomir</strong></font><br>
            School of Communications and Informatics,<br>
            Victoria University of Technology,<br>
            P.O Box 14428,<br>
            Melbourne City MC,<br>
            Victoria 8001, Australia <a
href="mailto:sever@matilda.vu.edu.au"><img src="images/mailto.gif" alt="Mail"
border="0" width="14" height="10"></a><br>
            URL: <a href="http://rgmia.vu.edu.au/SSDragomirWeb.html"
target="_top">http://rgmia.vu.edu.au/SSDragomirWeb.html</a><br>
            Telephone: +61 3 9688 4437<br>
            Facsimile: +61 3 9688 4050</td>
        </tr>
      </table>
      <blockquote>
        <p><strong>Managing Editor:</strong></p>
      </blockquote>
      <table border="0" cellpadding="0" cellspacing="0" width="100%"
bgcolor="#FFFFCC">
        <tr>
          <td width="20%"> </td>
          <td width="80%"><font color="#C60000"><strong>John
Roumeliotis</strong></font><br>
            School of Communications and Informatics,<br>
            Victoria University of Technology,<br>
            P.O. Box 14428,<br>
            Melbourne City MC,<br>
            Victoria 8001, Australia  <a
href="mailto:john.roumeliotis@vu.edu.au"><img src="images/mailto.gif" alt="Mail"
border="0" width="14" height="10"></a><br>
            URL: <a href="http://dingo.vu.edu.au/~johnr">http://dingo.vu.edu.au/~johnr</a><br>
            Telephone: +61 3 9688 4683<br>
            Facsimile: +61 3 9688 4050</td>
        </tr>
      </table>
      <blockquote>
        <p><strong>Assistant Editor:</strong></p>
      </blockquote>
      <table border="0" cellpadding="0" cellspacing="0" width="100%"
bgcolor="#FFFFCC">
        <tr>
          <td width="20%"> </td>
          <td width="80%"><font color="#C60000"><strong>Pui Ling
Pang</strong></font><br>
            School of Communications and Informatics,<br>
            Victoria University of Technology,<br>
            P.O. Box 14428,<br>
            Melbourne City MC,<br>
            Victoria 8001, Australia <a href="mailto:anna@sci.vu.edu.au"><img
src="images/mailto.gif" alt="Mail" border="0" width="14" height="10"></a><br>
            Telephone: +61 3 9688 4264<br>
            Facsimile: +61 3 9688 4908</td>
        </tr>
      </table>
       
      <blockquote>
        <p><strong><a name="office"></a>Editorial Office:</strong></p>
      </blockquote>
      <table border="0" cellpadding="0" cellspacing="0" width="100%"
bgcolor="#FFFFCC">
        <tr>
          <td width="20%"> </td>
          <td width="80%">Journal of Inequalities in Pure and Applied
            Mathematics,<br>
            C/O Ms. Pui Ling Pang,<br>
            School of Communications and Informatics,<br>
            Victoria University of Technology,<br>
            P.O. Box 14428,<br>
            Melbourne City MC,<br>
            Victoria 8001, Australia <a href="mailto:jipam@sci.vu.edu.au"><img
src="images/mailto.gif" alt="Mail" border="0" width="14" height="10"></a><a
href="mailto:anna@sci.vu.edu.au"><br>
            </a>Telephone: +61 3 9688 4264<br>
            Facsimile: +61 3 9688 4908</td>
        </tr>
      </table>
       
      <p>If you think our operation is deficient in some way, or equivalently,
      you have some suggestions for improvement then please don't hesitate to
      contact the Managing Editor (he would also welcome positive comments about
      how good our web site is!).</p>
      <p>For specific queries you could contact the chair responsible for each
      of our separate areas of operation. These are the <a
href="management.asp#board">Managerial
      Board</a>, <a href="management.asp#admin">Administration</a>, <a
href="management.asp#article">Article
      Design</a>, <a href="management.asp#publicity">Publicity</a> and <a
href="management.asp#web">Web
      Design</a>; an asterisk indicates the chair of the appropriate team.</p>
    </td>
  </tr>
</table>
```

```
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module ContactUs.asp

MODULE NAME – COPYRIGHT.ASP

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Copyright</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="COPYRIGHT"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
            <h1 align="center"><a name="Copyright"></a><font size="6"><font
color="#FF0000">C</font>opyright</font></h1>
            <p align="center"> </p>
<p align="left"><b><font size="4">    Transfer of Copyright</font></b></p>
<p align="left">    Upon acceptance of the manuscript ("the work") the author(s) will
transfer to
Victoria University of Technology (the publisher of "JIPAM") absolutely the 
copyright he or she may have in the work.</p>
<p align="left">The author(s) will be required to state that they own the copyright of
the work to the
extent that they can effectively give the above transfer free of any title, or other
claim, of 
any third person. </p>
<p align="left">The author(s) shall not publish the work in any other journal, whether
electronic 
or otherwise, in the same or substantially the same form as the work, without 
acknowledging prior publication in this journal. </p>
<p align="left">Authors must sign a Transfer of Copyright form.  You can obtain this
form by 
clicking <a href="common/copyright.pdf">here</a>.   </p>
<p align="left"> </p>
<p align="left"><b><font size="4">Rights of the author</font></b></p>
<p align="left">      Upon transfer of copyright to Victoria University of
Technology, publisher of JIPAM, the author retains the      following rights:</p>
<ol>
  <li>
    <p align="left">Patent and trademark rights and rights to any process or
procedure described in the work.</li>
  <li>
    <p align="left">The right to photocopy or make single electronic copies of
the work for their own personal use, including for their own
    classroom use, or for the personal use of colleagues,          provided the copies
are not offered for sale</li>
  <li>
    <p align="left">The right to <b> post the electronic version of their 
published work on one personal web
    site</b>.  This may          include the author's own single "home page"
    but not in any other systematic way outside the author's employing 
institution (e.g. via an email list or public file
    server).</li>
  <li>
    <p align="left">The right, subsequent to publication, to use the work or any 
    part thereof free of charge in a printed compilation of other  works of their
own, such as collected writings
    or lecture  notes, in a thesis, or to expand the work into a book-
length 
    form for publication.  </li>
</ol>
```

```
<p align="left">All copies, paper or electronic, or other use of the work must 
include an indication of the copyright ownership and a full  citation of the
journal source.  Please refer 
requests for all  uses not included above, including the authorization of
third 
parties to reproduce or otherwise use all or part of the work  (including figures
and tables) to: </p>
<p align="left">                        <a href="mailto:anna@sci.vu.edu.au">
Assistant   Editor </a><br>
Journal of  Inequalities in  Pure and  Applied 
Mathematics <br>
School of  Communications  and  Informatics (F019) <br>
Victoria                        University<br>
                        PO Box                                  14428, <br>
Melbourne                        City MC <br>
Victoria 8001, AUSTRALIA<br>
<br>
<br>
</p>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

APPENDIX C – MODULE COPYRIGHT.ASP

Module Name – Default.asp

```
<%@ Language=VBScript %>
<HTML>

<HEAD>
<TITLE>Welcome to JIPAM</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="MAIN"
%>
<!-- #include virtual="/jipam/common/common.inc" -->
<!-- #include file="common/Mainheader.inc" -->
<!-- #include file="common/menu.inc" -->
<I>Welcome to the first issue of the electronic journal</I>
<P ALIGN="center"><B><FONT SIZE="4"><I>Journal of Inequalities in Pure and
Applied Mathematics</I></FONT></B> <B><FONT SIZE="4"><I>(JIPAM)</I></FONT></B>
</P>
<P ALIGN="left"><I>Below we outline a few points worth noting:</I></P>
<UL>
<LI> <P ALIGN="left"><I>JIPAM is a peer-reviewed international journal in the
theory of mathematical inequalities and their applications.  The
<A HREF="management.asp">editorial board</A> is comprised of 49 internationally
recognized researchers - many of which are world leaders in the their
field.   <BR>
The journal was established by the VUT members of the Research Group in
Mathematical Inequalities and Applications (<A
HREF="http://rgmia.vu.edu.au">RGMIA</A>).  The phenomenal success of the
RGMIA (in 24 months the RGMIA has grown to over 240
<A HREF="http://rgmia.vu.edu.au/alpha_list_new.htm">members</A> worldwide and
published 11 issues of the <A
HREF="http://rgmia.vu.edu.au/reports.html">Research Report Collection</A>)
convinced us that a journal like JIPAM could succeed and indeed flourish.</I>
</P>
</LI>
<LI> <P ALIGN="left"><I>JIPAM is an electronic publication of the School of
Communications and Informatics, Victoria University of Technology.  Some
of the advantages of using the internet as the publishing medium are:</I> </P>
<UL>
<LI> <P ALIGN="left"><I>quick publication.  Our first call for papers was
in September, 1999 and in (less than) seven months, we've been able to receive
manuscripts, referee, typeset and publish.  Refereeing takes place in less
than 100 working days.  <B>A main aim of JIPAM is rapid dissemination of
results.</B></I></P>
</LI>
<LI> <P ALIGN="left"><I>cost.  Due to the low cost of electronic
publications, subscription rates (which will be free for this year) can be kept
to a minimum.</I></P>
</LI>
<LI> <P ALIGN="left"><I>flexible medium.</I>  <I>We can accept longer
review papers as we are not bound by page restrictions.  </I><I>In
addition, we further exploit the electronic nature of the journal by offering
two versions of each article - one suitable for printing and another designed
for viewing on screen.   See our <A HREF="pdf.asp">PDF</A> page for
more details. </I></P>
</LI>
</UL>
</LI>
<LI> <P ALIGN="left"><I>Publication will occur twice yearly in March and
October.</I></P>
</LI>
<LI> <P ALIGN="left"><I>To encourage submission of manuscripts of the highest
quality, JIPAM will offer an annual <A HREF="authors.asp#Prize">prize</A> to
the best article.</I></P>
</LI>
<LI> <P ALIGN="left"><I>This web site has been designed to be stable and quick
loading.  There is no Java or Javascript technology employed and each page
is less than 100k in size.  Once the top and side images are loaded,
navigation should be quick and easy.  For best viewing we recommend a
resolution of at least 800x600 with 256 or more colours.</I></P>
</LI>
</UL>
<P ALIGN="left"><I>Finally, we hope you enjoy reading the articles in this
first issue and we ask that you keep JIPAM in mind when considering your next
publication.</I></P>
<P ALIGN="right"><A HREF="http://rgmia.vu.edu.au/SSDragomirWeb.html">Sever S.
Dragomir</A><BR>
Editor-in-Chief<BR>
JIPAM </P>
<P ALIGN="right">24 April 2000</P>
<!-- #include file="common/Footer.inc" -->
```

Appendix C – Module Default.asp

MODULE NAME – DISPARTICLE.ASP

```
<%@ Language=VBScript %>
<html>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<head>
<title>Article Display for Jipam</title>
</head>
<%
  Dim PageName
  PageName="DISPLAYART"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->

<center>
<%
function ExpandAbstractString(InString)
  dim ststring
  dim Partstr
  dim start
  Partstr = InString
  do while len(Partstr) > 0
    start = Instr(1,Partstr,"##")
    Imend = Instr(start+2,PartStr,"##")
    if start = 0 or Imend=0 then 'transfer remainder of Partstr to ststring
      ststring = ststring + Partstr
      ExpandAbstractString = ststring
      exit do
    end if
    ststring = ststring + Mid(PartStr,1,start-1)
    AltStart = Instr(start+2,PartStr,"#")
    if AltStart <> Imend then
       Image = Mid(PartStr,start+2,AltStart -Start-2)
       AltText = Mid(PartStr,AltStart+1,Imend-AltStart-1)
    else
       Image = Mid(PartStr,start+2,Imend -Start-2)
       AltText = Image
    end if
    ImgStr = "<img border=""0"" align=""BOTTOM"" src=""ShowArtImg.asp?AI=" +
strArticleId
    ImgStr =ImgStr + "&IM=" + Mid(Image,6,2) + """ alt=""" + AltText + """ >"
    ststring = ststring + ImgStr
    PartStr = Mid(PartStr,Imend+2,len(PartStr))
  loop

end function
   Dim oCon
   Dim oCmd
   Dim strArticleId
   Dim strArticleTitle
   Dim strReceivedDate
   Dim strAcceptDate
   Dim intEditorId
   Dim strAbstract
   Dim strEmailAddress
   Dim strWebAddress
   Dim strSmallDesc
   Dim strAddressLine1
   Dim strAddressLine2
   Dim strAddressLine3
   Dim strAddressLine4
   Dim strAddressLine5
   Dim strEditorName
   Set Ors= nothing
   strArticleId=Request.QueryString("AI")
   if len(strArticleId) < 1 then strArticleId="000_00"
   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   oCon.Open strConnection

   'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = "DispArtDetails"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute(lngRecsAffected, Array(strArticleId, MaxLines))
   if oRs.BOF then
     Response.Write("The article requested is not yet
published</CENTER></BODY></HTML>")
     Response.End
   end if
   ' get article details
   strArticleTitle = oRs.Fields("ArticleTitle")
   strReceivedDate = oRs.Fields("ReceiveDate")
   strAcceptDate = oRs.Fields("AcceptDate")
   intEditorId = oRs.Fields("Editor_ID")
   strAbstract = ExpandAbstractString(oRs.Fields("Abstract"))
' need to expand the abstract for images
%>
<h2><%=strArticleTitle%></h2>
<p></p>
<table WIDTH="80%" BORDER="0">
<%
   oRs.Close
   Set oCmd1 = Server.CreateObject("ADODB.Command")
   oCmd1.ActiveConnection = oCon
   oCmd1.CommandType = 4           'stored procedure
   oCmd1.CommandText = "DispArtAuthors"   'procedure name
   Set oRs1 = oCmd1.Execute(lngRecsAffected, Array(strArticleId, MaxLines))
   strSmallAddress = Ors1.Fields("smallDesc")
   Do While NOT oRs1.Eof
     ' output author details
     if strSmallAddress <> Ors1.Fields("smallDesc") then 'this author not at same
instution
       Response.Write("<TR><TD></TD></TR><TR><TD></TD></TR>") ' blank line
       if len(strAddressLine1) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddressLine1) & "</CENTER></TD></TR>")
       if len(strAddressLine2) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine2) & "</CENTER></TD></TR>")
       if len(strAddressLine3) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine3) & "</CENTER></TD></TR>")
       if len(strAddressLine4) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine4) & "</CENTER></TD></TR>")
       if len(strAddressLine5) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine5) & "</CENTER></TD></TR>")
     end if
     strAddressLine1= oRs1.Fields("AddressLine1")
     strAddressLine2= oRs1.Fields("AddressLine2")
     strAddressLine3= oRs1.Fields("AddressLine3")
     strAddressLine4= oRs1.Fields("AddressLine4")
     strAddressLine5= oRs1.Fields("AddressLine5")
```

```
     Response.Write("<TR><TD><CENTER>" & oRs1.Fields("DisplayName") &
"</CENTER></TD></TR>")

     strEmailAddress = oRs1.Fields("EMailAddress")
     strWebAddress = oRs1.Fields("WebAddress")
     if len(strEMailAddress) > 0 then
        Response.Write("<TR><TD><CENTER>")
        Response.Write("E-Mail: <A HREF = mailto:" & strEmailAddress & ">" &
strEmailAddress & "</A>" )
        Response.Write("</CENTER></TD></TR>")
     end if
     if len(strWebAddress) > 0 then
        Response.Write("<TR><TD><CENTER>")
        Response.Write("URL: <A HREF = http://" & strWebAddress & ">" & strWebAddress
& "</A>")
        Response.Write("</CENTER></TD></TR>")
     end if
     strSmallAddress = Ors1.Fields("smallDesc")
     oRs1.MoveNext
   loop
   oRs1.Close
   Response.Write("<TR><TD></TD></TR>") ' blank line
   if len(strAddressLine1) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddressLine1) & "</CENTER></TD></TR>")
   if len(strAddressLine2) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine2) & "</CENTER></TD></TR>")
   if len(strAddressLine3) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine3) & "</CENTER></TD></TR>")
   if len(strAddressLine4) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine4) & "</CENTER></TD></TR>")
   if len(strAddressLine5) > 0 then Response.Write("<TR><TD><CENTER>" &
UCase(strAddRessLine5) & "</CENTER></TD></TR>")
   Set oRs1 = nothing
   Set oCmd1 = nothing

   'get editor Name
   Set oCmd2 = Server.CreateObject("ADODB.Command")
   oCmd2.ActiveConnection = oCon
   oCmd2.CommandType = 4           'stored procedure
   oCmd2.CommandText = "DispAuthor"   'procedure name
   Set oRs2 = oCmd2.Execute(lngRecsAffected, Array(intEditorId, MaxLines))
   strEditorName = oRs2.Fields("DisplayName")
   strEmailAddress = oRs2.Fields("EMailAddress")
   if len(strReceivedDate) > 0 or len(strAcceptDate) > 0 then
      Response.Write("<TR><TD><CENTER>")
      if len(strReceivedDate) > 0 then Response.Write("Received " &
FormatDateTime(strReceivedDate,1) & "; ")
      if len(strAcceptDate) > 0 then  Response.Write("Accepted " &
FormatDateTime(strAcceptDate,1) & ". ")
      Response.Write("</CENTER></TD></TR>")
   end if
   Response.Write("<TR><TD><CENTER>")
   Response.Write("Communicated by: <A HREF = mailto:" & strEmailAddress & ">" &
strEditorName & "</A>" )
   Response.Write("</CENTER></TD></TR></TABLE><P></P><HR>")
   oRs2.Close
   Set oRs2 = nothing
   Set oCmd2 = nothing
   Response.Write("<TABLE WIDTH=""80%""<TR><TD><B>ABSTRACT. </B>" & strAbstract &
"</TD></TR></TABLE><P></P>")

   Set oCmd3 = Server.CreateObject("ADODB.Command")
   oCmd3.ActiveConnection = oCon
   oCmd3.CommandType = 4           'stored procedure
   oCmd3.CommandText = "DispArtKeyWords"   'procedure name
   Set oRs3 = oCmd3.Execute(lngRecsAffected, Array(strArticleId, MaxLines))

   if NOT oRs3.BOF then
     strKeyWord = oRs3.Fields("KeyWord")
     Response.Write("<TABLE WIDTH=""60%""><TR><TD>KeyWords: <BR><I>" & strKeyWord)
     oRs3.MoveNext
     do while not oRs3.Eof
       strKeyWord = oRs3.Fields("KeyWord")
       Response.Write(", " & strKeyWord)
       oRs3.MoveNext
     loop
     Response.Write("</I></TD></TR></TABLE><P></P>")
   end if
   oRs3.Close
   set oRs3 = Nothing
   Set oCmd3 = nothing

   ' math codes
   Set oCmd4 = Server.CreateObject("ADODB.Command")
   oCmd4.ActiveConnection = oCon
   oCmd4.CommandType = 4           'stored procedure
   oCmd4.CommandText = "DispArtMathCodes"   'procedure name
   Set oRs4 = oCmd4.Execute(lngRecsAffected, Array(strArticleId, MaxLines))

   if NOT oRs4.BOF then
%>
<table WIDTH="60%">
<tr>
<td><a HREF="<%=MathCodeUrl%>">2000 Mathematics Subject Classification</a>:<br>
<%
     strMathCode = oRs4.Fields("MathCode")
     Response.Write(strMathCode)
     oRs4.MoveNext
     do while not oRs4.Eof
       strMathCode = oRs4.Fields("MathCode")
       Response.Write(", " & strMathCode)
       oRs4.MoveNext
     loop
     Response.Write("</TD></TR></TABLE><P></P>")
   end if
   oRs4.Close
   set oRs4 = Nothing
   Set oCmd4 = nothing

   ' Down Loads
   Set oCmd5 = Server.CreateObject("ADODB.Command")
   oCmd5.ActiveConnection = oCon
   oCmd5.CommandType = 4           'stored procedure
   oCmd5.CommandText = "DispArtDownloads"   'procedure name
   Set oRs5 = oCmd5.Execute(lngRecsAffected, Array(strArticleId, MaxLines))

   if NOT oRs5.BOF then
%>
      <hr>
      <p align="left"><b>Download this article :</b>
      <table border="0" width="100%" cellpadding="5">
        <tr>
          <td width="50%">
<%
   if  IsLoggedOn() then
    do while not oRs5.Eof
```

Appendix C – Module DispArticle.asp

```
      intPrintType = oRs5.Fields("PrintType")
      strPrintShortDesc = oRs5.Fields("PrintShortDesc")
      strImageFile = oRs5.Fields("ImageFile")
      strHref ="DOWNARTICLE.ASP?AI=" & strArticleId & "&PI=" & intPrintType

%>
            <p align="left">
            <%=strPrintShortDesc%>:        <a
href="<%=strHref%>" target="_blank"><img border="0" src="images/<%=strImageFile%>"
align="absmiddle" WIDTH="100" HEIGHT="93"></a>
 <%
      oRs5.MoveNext
    loop
    oRS5.Close
    Set ORs5 = nothing
    Set oCmd5 = nothing
    else
    ' not logged on
%>
<p align="left">To access the full text articles, you must <a HREF="login.asp">log
on</a>
 or <a HREF="subscribe.asp">subscribe</a> to this site.
<%
    end if

%>
          </td><td width="50%" bgcolor="#FFCCFF"><i>To view these files we recommend
            you save them to your file system and then view by using the Adobe
            Acrobat Reader. </i>
            <p><i>That is, click on the icon using the 2nd mouse button and
            select &quot;</i>Save Target As...&quot; (Microsoft Internet
            Explorer) <i>or</i> &quot;Save Link As...&quot; (Netscape
            Navigator).</p>
            <p>See our <a href="pdf.asp">PDF pages</a> for more information.</p>
          </td>
        </tr>
      </table>
<%
   end if
%>
</center>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module DispArticle.asp

Module Name – DispAuthTitle.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<title>Display Details of Author JIPAM</title>
</head>
<%
  Dim PageName
  PageName="DISPLAYAUTH"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->

<center>
<%
sub DisplayAnArticle()
%>
<LI> <P ALIGN="left"><%=strAuthorList%><BR>
<A HREF=DISPARTICLE.ASP?AI=<%=strArticleId%>><%=strArticleTitle%></A>
<%
     if len(strVolume) > 0 then
%>
, <A HREF=DISPVOLUME.ASP?VI=<%=intVolumeNo%>>Volume <%=strVolume%>, Issue
<%=strIssue%>, <%=strIssueYear%></A>.
<%
     else
     Response.Write(", Not yet published.")
     end if
%>
</P>
</LI>
<%
end sub
%>
<H1 ALIGN="center"><FONT
COLOR="#FF0000">P</FONT>ublication <FONT COLOR="#FF0000">L</FONT>ist
</H1>
<%

   Dim intUserId
   intUserId=Request.QueryString("UI")
   if len(intUserId) < 1 then intUserId=1 'this will fail as user_id start at 101

   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   oCon.Open strConnection

   'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = "DispAuthor"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute(lngRecsAffected, Array(intUserId, MaxLines))
   if oRs.EOF then
     Response.Write("<p>The requested author is not available.</p>")
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
     Response.End
   end if

   strDisplayName=Trim(oRs.Fields("DisplayName"))
   strName = Trim(oRs.Fields("FirstName")) & " " & Trim(oRs.Fields("OtherInits")) & "
" & Trim(oRs.Fields("Surname"))
   strAddress= Trim(Ors.Fields("AddressLine1"))
   strAddressl= Trim(Ors.Fields("AddressLine2"))
   if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddressl
   strAddressl= Trim(Ors.Fields("AddressLine3"))
   if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddressl
   strAddressl= Trim(Ors.Fields("AddressLine4"))
   if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddressl
   strAddressl= Trim(Ors.Fields("AddressLine5"))
   if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddressl
   strWebAddress=Trim(Ors.Fields("WebAddress"))
   strEmailAddress=Trim(Ors.Fields("EmailAddress"))
%>
<TABLE BORDER="0" WIDTH="100%" CELLSPACING="0" CELLPADDING="0">
<TR>
<TD WIDTH="100%"> <P ALIGN="left"><FONT SIZE="5">
<B><%=strName%><BR>
</B></FONT><%=strAddress%><BR>
<%
   if len(strEmailAddress) > 0 then
%>
EMail: <A HREF="mailto:<%=strEmailAddress%>"><%=strEmailAddress%></A>
<%
   end if
   if len(strWebAddress) > 0 then
%>
<BR>URL: <A HREF="http://<%=strWebAddress%>">http://<%=strWebAddress%></A>
<%
   end if
%>
</P>
<P></P>
<P></P>
<%
   Set oRs=nothing
   Set oCmd=nothing
   Set oCmdl = Server.CreateObject("ADODB.Command")
   oCmdl.ActiveConnection = oCon
   oCmdl.CommandType = 4           'stored procedure
   oCmdl.CommandText = "DispAuthArtList"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs1 = oCmdl.Execute(lngRecsAffected, Array(intUserId))
   if oRs1.EOF then
     Response.Write("<P>This author has no articles.</P>")
%>
</TD></TR></TABLE>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
     Response.End
   end if
   Response.Write("<UL>")
   strArticleId = Ors1.Fields("Article_Id")
   strAuthorList=""
   Do While NOT oRs1.Eof
     if strArticleId = Ors1.Fields("Article_Id") then ' another author
        if len(strAuthorList)> 0 then strAuthorList=strAuthorList & ", "
```

```
        if strDisplayName = Ors1.Fields("DisplayName") then 'the author being listed
           strAuthorList=strAuthorList & strDisplayName
        else
         strAuthorList=strAuthorList & "<A HREF=DISPAUTHTITLE.ASP?UI=" &
            Ors1.Fields("User_id") & ">" & Ors1.Fields("DisplayName") & "</A>"
        end if
      else ' a new record
        DisplayAnArticle
        if strDisplayName = Ors1.Fields("DisplayName") then 'the author being listed
           strAuthorList= strDisplayName
        else
           strAuthorList="<A HREF=DISPAUTHTITLE.ASP?UI=" &
              Ors1.Fields("User_id") & ">" & Ors1.Fields("DisplayName") & "</A>"
        end if
      end if
      strArticleId = Ors1.Fields("Article_Id")
      strArticleTitle=Ors1.Fields("ArticleTitle")
      intVolumeNo = Ors1.Fields("Volume_Id")
      strVolume = Ors1.Fields("Volume_NO")
      strIssue = Ors1.Fields("Issue_No")
      strIssueYear = Ors1.Fields("IssueYear")
      oRs1.MoveNext
    loop
    DisplayAnArticle
%>
</UL>
</TD>
</TR>
</TABLE>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module DispAuthTitle.asp

MODULE NAME – DISPVOLUME.ASP

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<title>Display articles in volume</title>
</head>
<%
  Dim PageName
  PageName="DISPLAYVOL"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->

<%
    Dim intVolumeId
    intVolumeId = Request.QueryString("VI")
    if intVolumeId < 1 or intVolumeId > LatestVolumeId then
       intVolumeId = LatestVolumeId
    end if
   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   oCon.Open strConnection

   'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = "DispVOLArticles"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute(lngRecsAffected, Array(intVolumeId, MaxLines))
   if oRs.BOF then 'this should never happen but ....
%>
<P>Sorry, the volume requested has not been published</P><P></P><HR>
<!-- #include virtual="/jipam/common/Footer.inc" -->

<%
     Response.End
   end if

%>
<TR>
<TD WIDTH="100%" VALIGN="top" ALIGN="left"> <H1 ALIGN="center"><FONT
COLOR="#FF0000">V</FONT>olume <%=oRs.Fields("Volume_NO")%>,
<FONT COLOR="#FF0000">I</FONT>ssue <%=oRs.Fields("Issue_NO")%>,
<%=oRs.Fields("IssueYear")%>
</H1>
<DIV ALIGN="center">
<TABLE BORDER="1" CELLPADDING="5" WIDTH="100%" CELLSPACING="15">
<TR>

<%
    intArticleNo=-1
    strAuthorList=""
    do while not oRs.Eof
      strAuthor = oRs.Fields("DisPlayName")
      intAuthorId = oRs.Fields("User_Id")

      if intArticleNo = oRs.Fields("ArticleNO") then 'just another author
        strAuthorList = strAuthorList & ", <A HREF=DISPAUTHTITLE.ASP?UI=" &
intAuthorId & ">" & strAuthor & "</A>"

      else ' new article
         if len(strAuthorList) then ' complete the previous article
%>
<P ALIGN="center"><%=strAuthorList%></P>
</TD>
</TR>
<%
         end if
        intArticleNo = oRs.Fields("ArticleNO")
        strArticleTitle = oRs.Fields("ArticleTitle")
        strArticleId = oRs.Fields("Article_ID")
%>
<TR>
<TD WIDTH="5%" VALIGN="top" ALIGN="center" BGCOLOR="#FFFFCC"><%=intArticleNo%>.</TD>
<TD WIDTH="95%" ALIGN="center" BGCOLOR="#FFFFCC"> <P ALIGN="center">
<A HREF="DISPARTICLE.ASP?AI=<%=strArticleId%>"><B><I><FONT COLOR="#000080">
<%=strArticleTitle%></FONT></I></B></A></P>
<%
        strAuthorList = "<A HREF=DISPAUTHTITLE.ASP?UI=" & intAuthorId & ">" &
strAuthor & "</A>"
      end if ' of new article
      oRs.MoveNext
    loop
    Set oRs=nothing
    Set oCmd=nothing
    ' complete author list
%>
<P ALIGN="center"><%=strAuthorList%></P>
</TD>
</TR>
</TABLE>
</DIV>
<P></P>
<P></P>
<%
   ' remove comment on next if-end block for correct processing
   ' and delete next assignment statement
   intVolumeId=2
   'if LatestVolumeId > 1 then
%>
<HR>
<DIV ALIGN="center">
<NOBR><B>Other Volumes:</B> </NOBR>
<%
   Set oCmd1 = Server.CreateObject("ADODB.Command")
   oCmd1.ActiveConnection = oCon
   oCmd1.CommandType = 4           'stored procedure
   oCmd1.CommandText = "VOLumeList"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result

   Set oRs1 = oCmd1.Execute(lngRecsAffected, Array(intVolumeId))
   Do While NOT  oRs1.Eof
     intVolumeNo = Ors1.Fields("Volume_Id")
     strVolume = "V" & Ors1.Fields("Volume_NO")& "." & Ors1.Fields("Issue_NO") & "." &
Ors1.Fields("IssueYear")
%>
```

APPENDIX C – MODULE DISPVOLUME.ASP

```
 <NOBR>[ <A
HREF=DISPVOLUME.ASP?VI=<%=intVolumeNo%>><%=strVolume%></A> ]</NOBR>
<%
     oRs1.MoveNext
    loop
    Set oRs1=nothing
    Set oCmd1=nothing
%>
</DIV>
<%
  'end if
%>
</TD>
</TR>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module DispVolume.asp

MODULE NAME – DOSEARCH.ASP

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<title>Perform Search or JIPAM</title>
</head>
<%
  Dim PageName
  PageName="DOSEARCH"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
  Dim strSearch
  Dim intOption
  Dim strSearchA
  Dim strSelOption
  Dim strExecOption
  Dim lngRecsAffected

  strSearch=Request.Form("TxtSearch")
  intOption =Request.Form("RdoSearch")
  if Len(strSearch) < 1 then
     strSearch = "ABCD"
  end if
  if intOption < 0 or intOption > 5 then
     intOption=1
  end if
  strSearchA = "%"+strSearch+"%"
  Select Case intOption
    Case 1
      strSelOption="Authors"
      strExecOption="SearchAuthors"
    Case 2
      strSelOption="Title"
      strExecOption="SearchTitles"
    Case 3
      strSelOption="Keyword"
      strExecOption="SearchKeyWords"
    Case 4
      strSelOption="Math Code"
      strExecOption="SearchMathCode"
    Case 5
      strSelOption="Abstracts"
      strExecOption="SearchAbstracts"
    Case Else
      intOption=1
      strSelOption="Authors"
      strExecOption="SearchAuthors"

  end Select

%>
<H1 ALIGN="center">
<FONT COLOR="#FF0000">S</FONT>earch <FONT COLOR="#FF0000">R</FONT>esults
</H1>
<%

   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   oCon.Open strConnection

   'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = strExecOption   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute(lngRecsAffected, Array(strSearchA, MaxLines))
   if ORs.Bof then
%>
<p>There were no records found within the JIPAM database that matched your search
criteria.</p>
<p>Change the form below if want to try again.</p>
<!-- #include virtual="/jipam/common/search.inc" -->
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
   Response.End
   end if
%>
<p>The following matched your search criteria</p>
<p></p>
<table WIDTH="95%" BORDER="1" CELLPADDING="2">
<tr>
<td WIDTH="225"> Article Title</td>
<td WIDTH="149"> Authors</td>
<td WIDTH="140"> Journal No</td>
</tr>
<%
   strArticleId=Ors.Fields("Article_Id")
   strAuthorList=""
   strVolume=""
   strArticleTitle=""
   Do While NOT oRs.Eof
     strAuthor = Ors.Fields("DisplayName")
     intAuthorId = Ors.Fields("User_ID")
     if strArticleId = Ors.Fields("Article_Id") then 'Just another author
        if len(strAuthorList) > 0 then strAuthorList = strAuthorList & ",  "
        strAuthorList = strAuthorList & "<A HREF=DISPAUTHTITLE.ASP?UI=" & intAuthorId
& ">" & strAuthor &"</A>"
     else  ' a complete entry entry has been formed
%>
<TR>
<TD WIDTH="70%"> <%=strArticleTitle%></TD>
<TD WIDTH="20%"> <%=strAuthorList%></TD>
<TD WIDTH="10%"> <%=strVolume%></TD>
</TR>
<%
        strAuthorList = "<A HREF=DISPAUTHTITLE.ASP?UI=" & intAuthorId & ">" &
strAuthor &"</A>"
     end if
     strArticleId = Ors.Fields("Article_Id")
     strArticleTitle="<A HREF=DISPARTICLE.ASP?AI=" & strArticleId & ">" &
Ors.Fields("ArticleTitle") &"</A>"
     intVolumeNo = Ors.Fields("Volume_Id")
     strVolume = "V" & Ors.Fields("Volume_NO")
     if len(strVolume) = 1 then
       strVolume=""
     else
       strVolume = strVolume & "/" & Ors.Fields("Issue_NO") & "/" &
Ors.Fields("IssueYear")
```

```
        strVolume="<A HREF=DISPVOLUME.ASP?VI=" & intVolumeNo & ">" & strVolume &"</A>"
      end if
    Ors.MoveNext
    loop
    ' complete the outstanding record
%>
<TR>
<TD WIDTH="70%"> <%=strArticleTitle%></TD>
<TD WIDTH="20%"> <%=strAuthorList%></TD>
<TD WIDTH="10%"> <%=strVolume%></TD>
</TR>
</table>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module DoSearch.asp

MODULE NAME – DOWNARTICLE.ASP

```
<%@ Language=VBScript %>
<% 'Response.buffer=true
%>
<!-- #include virtual="/jipam/common/common.inc" -->
<!-- #include virtual="/jipam/common/error.inc" -->
<% ' this process returns a binary file
  Dim PageName
  Dim strArticleId
  Dim intPrintType
  PageName="NONAME"
Sub TermMess(Mess)
%>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<HEAD>
<TITLE>Down Map</TITLE>
</HEAD>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<I><%=Mess%></I>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
     Response.End
end sub

  strArticleId = Request.QueryString("AI")
  intPrintId = Request.QueryString("PI")
  if not IsLoggedOn() or len(strArticleId) <> 6 or intPrintId < 1 then
     TermMess "Sorry,you cannot download the requested article because you have not
logged on."
     Response.End
  end if
  On Error Resume Next
   Err.Clear()
   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   if Err.number > 0 then TermMess "Unable to connect to database(1)"
   Err.Clear ()
   oCon.Open strConnection
   if Err.number > 0 then TermMess "Unable to connect to database(2)"
  'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   if Err.number > 0 then TermMess "Unable to connect to database(3)"
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = "GetArticleFull"   'procedure name
   lngRecsAffected=0
   arrParams=Array(strArticleId,  intPrintId)
   'execute the command, supplying the parameters, and get the result
   Err.Clear()
   Set oRs = oCmd.Execute(lngRecsAffected,arrParams )
   if Err.number > 0 then TermMess "Unable to connect to database(4)"
   if oRs.Eof then
      TermMess "Sorry, the requested article is not available for download."
     Response.End
   end if

   strHtmlResponse=oRs.Fields("HtmlDescription")
   Response.ContentType = strHtmlResponse

   'Response.Flush()
   Set adoFldBlob=oRs("TheArticle")
   lngFieldDataLength=adoFldBlob.ActualSize

   lngBlockCount=lngFieldDataLength / 4096
   lngRemainingData=lngFieldDataLength mod 4096

   'Response.Buffer =  false
   for lngCounter=1 to lngBlockCount
      Response.BinaryWrite (adoFldBlob.GetChunk(4096))
   next
   if lngRemainingData > 0 then
       Response.BinaryWrite (adoFldBlob.GetChunk(lngRemainingData))
   end if
   adoFldBlob.Close
   set adoFldBlob =nothing
   set ors=nothing
   set oCmd=nothing
   Response.End
%>
```

Module Name – Editors.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Editors</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="EDITORS"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<TABLE WIDTH="100%">
<TR>
<TD> <H1 ALIGN="center"><FONT
COLOR="#FF0000">F</FONT>oundation <FONT COLOR="#FF0000">E</FONT>ditors</H1>
</TD>
</TR>
</TABLE>
<TABLE WIDTH="100%">
<%
sub DisplayEditor( oRsIn)
  if oRsIn.Eof then
%>
<TD WIDTH="49%" STYLE="text-align: left; vertical-align: text-top" VALIGN="top">
</TD>
<%
     exit sub
  end if
'    An Editor Record
     strEdName = Trim(oRsIn.Fields("FirstName")) & " " &
Trim(oRsIn.Fields("OtherInits")) & " " & Trim(oRsIn.Fields("Surname"))
     strAddress= Trim(OrsIn.Fields("AddressLine1"))
     strAddress1= Trim(OrsIn.Fields("AddressLine2"))
     if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddress1
     strAddress1= Trim(OrsIn.Fields("AddressLine3"))
     if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddress1
     strAddress1= Trim(OrsIn.Fields("AddressLine4"))
     if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddress1
     strAddress1= Trim(OrsIn.Fields("AddressLine5"))
     if len(strAddress1) > 0 then strAddress = strAddress & "<BR>" & strAddress1
     strWebAddress=Trim(OrsIn.Fields("WebAddress"))
     strEmailAddress=Trim(OrsIn.Fields("EmailAddress"))
     strFoundation=""
     if trim(OrsIn.Fields("FoundingFlag")) = "Y" then strFoundation=" (Fd)"
%>
<TD WIDTH="49%" STYLE="text-align: left; vertical-align: text-top" VALIGN="top">
<UL><LI>
<STRONG><FONT
COLOR="#C60000"><%=strEdName%> <%=strFoundation%><BR></FONT></STRONG>
<FONT SIZE="2"><%=strAddress%>
</FONT> 
<A HREF="mailto:<%=strEmailAddress%>"><IMG SRC="/jipam/images/mailto.gif"
ALT="Email this editor" BORDER="0" WIDTH="14" HEIGHT="10"></A> 
<%
   if len(strWebAddress) > 0 then
%>
<A HREF="http://<%=strWebAddress%>" TARGET="_top"><IMG SRC="/jipam/images/globe.jpg"
   BORDER="0" ALT="Link to this editor's web page"
ALIGN="absmiddle" WIDTH="25" HEIGHT="25"></A>
<%
   end if
%>
</LI></UL>
</TD>
<%
  oRsIn.MoveNext
end sub
   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   oCon.Open strConnection

   'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4          'stored procedure
   oCmd.CommandText = "FullEditorList"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute( lngRecsAffected)
   do while not oRs.Eof
     Response.Write("<TR>")
     DisplayEditor Ors
%>
<TD WIDTH="2%" STYLE="text-align: left; vertical-align: text-top" VALIGN="top"></TD>
<%
     DisplayEditor Ors
     Response.Write("</TR>")
   loop
   set oRs=nothing
   set oCmd=nothing
%>
</TABLE>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

MODULE NAME – FEEDBACK.ASP

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Feedback</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="FEEDBACK"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<h1 align="center"><font color="#FF0000">F</font>eedback</h1>
<%
sub TermMess( theMess)
%>
<%=theMess%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
   Response.end
end sub

Dim iMessageType

  strMessageType=Request.Form.Item("MessageType")
  if len(strMessageType) > 0 then ' we are processing a message
Dim strSubject
Dim strSubjectOther
Dim strComments
Dim strUserName
Dim strUserEmail
Dim strUserTel
Dim strUserFax
Dim strContactReq
Dim strAdmin
  strSubject=Request.Form.Item("Subject")
  strSubjectOther=Request.Form.Item("SubjectOther")
  if len(strSubjectOther) < 1 then strSubjectOther=""
  strComments=Request.Form.Item("Comments")
  if len(strComments) < 1 then strComments=""
  strUserName=Request.Form.Item("Username")
  if len(strUserName) < 1 then strUserName=""
  strUserEmail=Request.Form.Item("UserEmail")
  if len(strUserEmail) < 1 then strUserEmail=""
  strUserTel=Request.Form.Item("UserTel")
  if len(strUserTel) < 1 then strUserTel=""
  strUserFax=Request.Form.Item("UserFAX")
  if len(strUserFax) < 1 then strUserFax=""
  strContactReq=Request.Form.Item("ContactRequested")
  if len(strContactReq) < 1 then strContactReq=""
  strAdmin =Application.Contents.Item("EmailAdmin")
  if len(strAdmin) < 1 then strAdmin=""
    Set oCon = Server.CreateObject("ADODB.Connection")
    oCon.Open strConnection

   'create a command and set the properties
    Set oCmd = Server.CreateObject("ADODB.Command")
    oCmd.ActiveConnection = oCon
    oCmd.CommandType = 4           'stored procedure
    oCmd.CommandText = "AddFeedback"   'procedure name
    lngRecsAffected=0
    arrParams=Array(strMessageType,
strSubject,strSubjectOther,strUserName,strUserEmail, _
       strUserTel,strUserFax,strContactReq,strComments,strAdmin)
   'execute the command, supplying the parameters, and get the result
   if len(strComments) > 0 then
      On Error Resume Next
      Set oRs = oCmd.Execute(lngRecsAffected,arrParams )
   end if
    TermMess "<p>Thank you for your feedback</p>"
  end if

%>

<p>Tell us what you think about our web site, our journal, or anything else that
comes to mind. We welcome all of your comments and suggestions, the only thing
we ask is that supply a name (and an email address if you would like a
response).</p>
<form method="POST" action="Feedback.ASP">
  <p><strong>What kind of comment would you like to send?</strong></p>
  <dl>
    <dd><input type="radio" name="MessageType" value="Complaint">Complaint <input
type="radio" name="MessageType" value="Problem">Problem
      <input type="radio" checked name="MessageType" value="Suggestion">Suggestion
      <input type="radio" name="MessageType" value="Praise">Praise</dd>
  </dl>
  <p><strong>What about us do you want to comment on?</strong></p>
  <dl>
    <dd><select name="Subject" size="1">
        <option selected>Web Site</option>
        <option>Journal</option>
        <option>Papers</option>
        <option>(Other)</option>
      </select> Other: <input type="text" size="26" maxlength="256"
name="SubjectOther"></dd>
  </dl>
  <p><strong>Enter your comments in the space provided below:</strong></p>
  <dl>
    <dd><textarea name="Comments" rows="5" cols="42"></textarea></dd>
  </dl>
  <p><strong>Tell us how to get in touch with you:</strong></p>
  <dl>
    <dd>
      <table>
        <tr>
          <td>Name
          <td><input type="text" size="35" name="Username">
        </tr>
        <tr>
          <td>E-mail
          <td><input type="text" size="35" maxlength="256" name="UserEmail">
        </tr>
        <tr>
          <td>Tel
          <td><input type="text" size="35" maxlength="256" name="UserTel">
        </tr>
        <tr>
          <td>FAX
          <td><input type="text" size="35" maxlength="256" name="UserFax">
        </tr>
      </table>
    </dd>
```

APPENDIX C – MODULE FEEDBACK.ASP

```
  </dl>
  <dl>
    <dd><input type="checkbox" name="ContactRequested" value="ContactRequested">
      Please contact me as soon as possible regarding this matter.</dd>
  </dl>
  <p><input type="submit" value="Submit Comments"> <input type="reset" value="Clear
Form"></p>
</form>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module Feedback.asp

MODULE NAME – FOOTER.INC

```
<!--after contents-->
</TD>
</TR>
<TR>
<TD><BR>
<BR>
<HR>
<CENTER>
<!--before copyright-->
 <P ALIGN="left"><EM>&copy;</EM> 2000 <A HREF="http://sci.vu.edu.au">School
of Communications and Informatics</A>, <A HREF="http://www.vu.edu.au">Victoria
University of Technology</A>. All rights reserved.<BR>
JIPAM is published by the School of Communications and Informatics which is
part of the <A HREF="http://www.vu.edu.au/foes/">Faculty of Engineering and
Science</A>, located in Melbourne, Australia. All correspondence should be
directed to the <A HREF="contactus.asp">editorial office</A>.</P>
<P ALIGN="left"><A HREF="copyright.asp">Copyright/Disclaimer</A></P
<!--after copyright-->
</CENTER>
</TD>
</TR>
<TR>
<TD><BR>
<HR>
<CENTER>
<!--before bottom menu-->
<%  if PageName<>"MAIN" then %>
<NOBR>[ <A HREF="/jipam/default.asp">Home </A>]</NOBR>
<% else %>
<NOBR>[ Home ]</NOBR>
<%
  end if
  if PageName<>"SCOPE" then %>
<NOBR>[ <A HREF="aimsscope.asp">About Jipam </A>]</NOBR>
<% else %>
<NOBR>[ About Jipam ]</NOBR>
<%
  end if
  if PageName<>"COPYRIGHT" then
%>
<NOBR>[ <A HREF="copyright.asp">Copyright </A>]</NOBR>
<% else %>
<NOBR>[ Copyright ]</NOBR>
<%
  end if
  if PageName<>"EDITORS" then
%>
<NOBR>[ <A HREF="Editors.asp">Editors </A>]</NOBR>
<% else %>
<NOBR>[ Editors ]</NOBR>
<%
   end if
  if PageName<>"WHATSNEW" then
%>
<NOBR>[ <A HREF="WhatsNew.asp">What's New </A>]</NOBR>
<% else %>
<NOBR>[ What's New ]</NOBR>
<%
  end if
  if PageName<>"DISPLAYVOL" then
%>
<NOBR>[ <A HREF="DispVolume.asp">Current Issue </A>]</NOBR>
<% else %>
<NOBR>[ Current Issue ]</NOBR>
<%
  end if
  if PageName<>"ACCEPTED" then
%>
<NOBR>[ <A HREF="accepted.asp">New Papers </A>]</NOBR>
<% else %>
<NOBR>[ New Papers ]</NOBR>
<%
  end if
  if PageName<>"SEARCH" then
%>
<NOBR>[ <A HREF="search.asp">Search</A> ]</NOBR>
<% else %>
<NOBR>[ Search</A> ]</NOBR>

<% end if
  if not ( PageName = "SUBSCRIBE" or PageName="LOGON" or Session("UserID")>0 or
Session("Locked")) then
%>
<NOBR>[ <A HREF="subscribe.asp">Subscribe </A> ]</NOBR>
<NOBR>[ <A HREF="login.asp">Login </A> ]</NOBR>
<% end if
  if PageName<>"AUTHORS" then
%>
<NOBR>[ <A HREF="authors.asp">Authors Info ]</NOBR>
<NOBR>[ <A HREF="authors.asp#Prize">Prize ]</NOBR>
<% else %>
<NOBR>[ Authors Info ]</NOBR>
<NOBR>[ Prize ]</NOBR>
<%  end if
  if PageName<>"FEEDBACK" then
%>
<NOBR>[ <A HREF="feedback.asp">Feedback</A> ]</NOBR>
<%  else %>
<NOBR>[ Feedback ]</NOBR>
<%
  end if
  if PageName<>"CONTACT" then
%>
<NOBR>[ <A HREF="contactus.asp">Contact Us</A> ]</NOBR>
<% else %>
<NOBR>[ Contact Us ]</NOBR>
<% end if %>
<%  if IsAdmin() then %>
<%  if PageName<>"TRAFFIC" then %>
<NOBR>[ <A HREF="TrafficReports.asp">Visitor Reports</A> ]</NOBR>
<% else %>
<NOBR>[ Traffic Reports ]</NOBR>
<% end if %>

<%  end if  %>

<!--after bottom menu-->
</CENTER>
</TD>
</TR>
</TABLE>
</TD>
</TR>
```

APPENDIX C – MODULE FOOTER.INC

```
</TABLE>
</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Appendix C – Module Footer.inc

MODULE NAME – GLOBAL.ASA

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>

'You can add special event handlers in this file that will get run automatically when
'special Active Server Pages events occur. To create these handlers, just create a
'subroutine with a name from the list below that corresponds to the event you want to
'use. For example, to create an event handler for Session_OnStart, you would put the
'following code into this file (without the comments):

'Sub Session_OnStart
'**Put your code here **
'End Sub

'EventName               Description
'Session_OnStart         Runs the first time a user runs any page in your application
'Session_OnEnd           Runs when a user's session times out or quits your application
'Application_OnStart     Runs once when the first page of your application is run for
the first time by any user
'Application_OnEnd       Runs once when the web server shuts down
Sub Application_onStart()
   On Error Resume Next
   strSQL = "INSERT INTO Sessions (EventDateTime,EventType) VALUES
(GetDate(),'Application Start')"
   WriteSessionData strSQL

   Application.Contents.Item("EmailAdmin")="EmailAdmin"

Application.Contents.Item("strConnection")="DSN=JIPAM;UID=Internet;PWD=internet;DATABA
SE=JIPAM2"

Application.Contents.Item("strConnLogs")="DSN=JIPAM;UID=JipamLog;PWD=JipamLog;DATABASE
=JIPAM2"
   Application.Contents.Item("MaxLines")=30
   Application.Contents.Item("MathCodeUrl")="http://www.ams.org/msc"
   Application.Contents.Item("LatestVolumId")=1
   Application.Contents.Item("LoginAttemps")=4
End Sub

Sub Session_onStart()
   'The variables to indicate a successful logon
   Dim UserLogon
   Dim UserID
   Dim UserRole
   Dim Locked
   Dim UsePass
   Dim LoginCount
   UserRole=0
   Locked=false
   UserId=0
   UsePass=""
   LoginCount=Application.Contents.Item("LoginAttemps")


   On Error Resume Next
   strSQL = "INSERT INTO Sessions " _
          &
"(EventDateTime,EventType,URL,Referer,RemoteHost,UserAgent,UALanguage,UserID,HostIP)"
_
          & " VALUES (GetDate(),'New Session', '" _
          & Request.ServerVariables("URL") & "', '" _
          & Request.ServerVariables("HTTP_REFERER") & "', '" _
          & Request.ServerVariables("REMOTE_HOST") & "', '" _
          & Request.ServerVariables("HTTP_USER_AGENT") & "', '" _
          & Request.ServerVariables("HTTP_ACCEPT_LANGUAGE") & "', " _
          & CLng(Session.SessionID) & ", '" _
          & Request.ServerVariables("LOCAL_ADDR") & "')"
   WriteSessionData strSQL
End Sub

Sub Application_onEnd()
   On Error Resume Next
   strSQL = "INSERT INTO Sessions (EventDateTime,EventType) VALUES
(GetDate(),'Application End')"
   WriteSessionData strSQL
End Sub

Sub Session_onEnd()
   On Error Resume Next
   strSQL = "INSERT INTO Sessions (GetDate(),EventDateTime,EventType, UserID) " _
       & "VALUES ('Session End', " & CLng(Session.SessionID) & ")"
   WriteSessionData strSQL
End Sub

Sub WriteSessionData(strSQL)
   On Error Resume Next
   Set oConn = Server.CreateObject("ADODB.Connection")
   oConn.open "DSN=JIPAM;UID=JipamLog;PWD=JipamLog;"
   oConn.Execute strSQL
   Set oConn = Nothing
End Sub

</SCRIPT>
```

MODULE NAME – GRAPHICLANG.ASP

```
<%@LANGUAGE="VBScript"%>
<% Server.ScriptTimeOut = 600 %>
<!-- #include virtual="/jipam/common/common.inc" -->
<%
Dim arrTopOnes(8, 1)

blnIE4 = False
strUA = Request.ServerVariables("HTTP_USER_AGENT")
intMSIE = Instr(strUA, "MSIE ") + 5
If intMSIE > 5 Then
  If CInt(Mid(strUA, intMSIE, Instr(intMSIE, strUA, ".") - intMSIE)) >= 4 Then
    blnIE4 = True
  End If
End If
%>

<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<HEAD>
<TITLE>Results of your traffic query</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="TRAFFICLAN"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
   if Not IsAdmin() then
      Response.Write("<B>Sorry, this function is not available </B>")
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
    Response.End
   end if
%>
<H1 ALIGN="center">
<FONT COLOR="#FF0000">G</FONT>raphical <FONT COLOR="#FF0000">B</FONT>rowser
<FONT COLOR="#FF0000">L</FONT>anguages
</H1>
<%
QUOT = Chr(34)
strWhere = " WHERE " & Request.Form("criteria")
strDisplay = Request.Form("display")
strSQL = "SELECT LangCount=SUM(ItemCount), Lang=MAX(LangText) " _
       & "FROM LangCodes INNER JOIN CountrySummary " _
       & "ON LangCodes.LangCode = CountrySummary.ItemText" & strWhere _
       & " GROUP BY LangText ORDER BY SUM(ItemCount) DESC"

 %>
<SPAN>Breakdown of browser languages for visitors to our
site<%=strDisplay%>:</SPAN><BR><P>

<%
On Error Resume Next
Set oConn = Server.CreateObject("ADODB.Connection")  'create a connection
Set oRs = Server.CreateObject("ADODB.Recordset")     'create a recordset
oConn.Open strConnLogs              'open connection using value in include file
oRs.Open strSQL, oConn, 3,, 1  'open recordset using adOpenStatic, , adCmdtext
If (oRs.EOF) Or (Err.Number <> 0) Then
  if oRs.EOF then
    Response.Write "<B>The query returned no records</B>"
  else
    Response.Write "<B>The database cannot be accessed</B>"
  end if
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
  Response.End
End If
intTTPos = 0
lngTotal = 0
'--get the top eight entries--
Do While Not oRs.EOF
  lngTotal = lngTotal + CLng(oRs("LangCount"))
  If intTTPos < 8 Then
    arrTopOnes(intTTPos, 0) = oRs("LangCount")
    arrTopOnes(intTTPos, 1) = oRs("Lang")
  End If
  intTTPos = intTTPos + 1
  oRs.MoveNext
Loop
lngRest = lngTotal
For intLoop = 0 To 7
  lngRest = lngRest - arrTopOnes(intLoop, 0)
Next
'just in case:
If lngRest < 0 Then lngRest = 0
arrTopOnes(8, 0) = lngRest
arrTopOnes(8, 1) = "Others"


'*** draw chart if is IE4 or better **
If blnIE4 Then
iDegPos = 0
iRectTop = -110
%>
<TABLE STYLE="height:400; width:700" >
<TR><TD>
<OBJECT ID=sgPie STYLE="position:relative; height:230; width:300; top:10; left:10"
  CLASSID="CLSID:369303C2-D7AC-11D0-89D5-00A0C90833E6">
<PARAM NAME="Line0001" VALUE="SetLineColor(0, 0, 0)">
<PARAM NAME="Line0002" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0003" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0004" VALUE="Oval(-130, -90, 200, 200, 0)">

<PARAM NAME="Line0005" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0006" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0007" VALUE="SetFillColor(215, 40, 40)">
<PARAM NAME="Line0008" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(arrTopOnes(0, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0009" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(0, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0010" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0011" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0012" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0013" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0014" VALUE="SetFillColor(255, 128, 0)">
<PARAM NAME="Line0015" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(arrTopOnes(1, 0) / lngTotal*360)) %>, 0)">
```

```
<PARAM NAME="Line0016" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(1, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0017" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0018" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0019" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0020" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0021" VALUE="SetFillColor(240, 240, 0)">
<PARAM NAME="Line0022" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(arrTopOnes(2, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0023" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(2, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0024" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0025" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0026" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0027" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0028" VALUE="SetFillColor(30, 200, 47)">
<PARAM NAME="Line0029" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(arrTopOnes(3, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0030" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(3, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0031" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0032" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0033" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0034" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0035" VALUE="SetFillColor(30, 203, 203)">
<PARAM NAME="Line0036" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(arrTopOnes(4, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0037" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(4, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0038" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0039" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0040" VALUE="Rect(100, <%= CStr(iRectTop) +5 %>, 20, 15, 0)">
<PARAM NAME="Line0041" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0042" VALUE="SetFillColor(80, 80, 218)">
<PARAM NAME="Line0043" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(arrTopOnes(5, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0044" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(5, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0045" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0046" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0047" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0048" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0049" VALUE="SetFillColor(190, 55, 190)">
<PARAM NAME="Line0050" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(arrTopOnes(6, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0051" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(6, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
```

```
<PARAM NAME="Line0052" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0053" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0054" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0055" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0056" VALUE="SetFillColor(128, 128, 0)">
<PARAM NAME="Line0057" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Cint(arrTopOnes(7, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0058" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(arrTopOnes(7, 0) / lngTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0059" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0060" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0061" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0062" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0063" VALUE="SetFillColor(0, 0, 0)">
<PARAM NAME="Line0064" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Cint(arrTopOnes(8, 0) / lngTotal * 360)) %>, 0)">
<PARAM NAME="Line0065" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
</OBJECT>
</TD><TD WIDTH=5></TD><TD>
<TABLE  CELLSPACING=7>
<%
For intLoop = 0 To 8
%>
<TR><TD ALIGN="LEFT">
<B><%= arrTopOnes(intLoop, 1) %></B>    Count: <B><%= arrTopOnes(intLoop, 0)
%></B>  (<B><%= FormatPercent(arrTopOnes(intLoop, 0) / lngTotal, 1) %></B>)
</TD></TR>
<%
Next
%>
</TD></TR></TABLE>
</TD></TR></TABLE>
<%
End If
%>
<TABLE WIDTH=100%>
<TR><TD ALIGN=CENTER>
The browser does not indicate the country that it is in when it requests a page,
and because of the indiscriminate way that the traditional county codes are used in
URLs
(i.e. sites that have .com in the URL are often not in the US, etc.), the only way to
reliably identify the browser is via the language code that it supplies. This
indicates
the language that the browser is set up for, and not necessarily the country it is
being used in.
However this does provide a fairly reliable guide to the locations of our visitors.
Here's the full list:<P>
</TD></TR></TABLE>
<CENTER><TABLE>

<TR>
<% For intCol = 1 To 3 %>
<TD NOWRAP ALIGN="RIGHT"><B>Count</B></TD><TD NOWRAP><B>  Language</B></TD><TD
NOWRAP>     </TD>
<% Next %>
</TR>

<% oRs.MoveFirst
Do While Not oRs.EOF
```

Appendix C – Module GraphicLang.asp

```
  Response.Write "<TR>"
  For intCol = 1 To 3
    If oRs.EOF Then
      Response.Write "<TD> </TD><TD> </TD><TD> </TD>"
    Else
      Response.Write "<TD NOWRAP ALIGN=RIGHT>" & oRs("LangCount") & "</TD><TD
NOWRAP>  " & oRs("Lang") & "  </TD><TD NOWRAP> </TD>"
    End If
    oRs.MoveNext
  Next
  Response.Write "</TR>" & vbCrlf
Loop
Set oConn = Nothing
%>
</TABLE></CENTER><BR>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module GraphicLang.asp

Module Name – GraphicOpSys.asp

```
<%@LANGUAGE="VBScript"%>
<% Server.ScriptTimeOut = 600 %>
<!-- #include virtual="/jipam/common/common.inc" -->
<%
blnIE4 = False
strUA = Request.ServerVariables("HTTP_USER_AGENT")
intMSIE = Instr(strUA, "MSIE ") + 5
If intMSIE > 5 Then
  If CInt(Mid(strUA, intMSIE, Instr(intMSIE, strUA, ".") - intMSIE)) >= 4 Then
    blnIE4 = True
  End If
End If
%>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<HEAD>
<TITLE>Results of your traffic query</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="TRAFFICOPS"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
   if Not IsAdmin() then
      Response.Write("<B>Sorry, this function is not available </B>")
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
    Response.End
   end if
%>
<H1 ALIGN="center">
<FONT COLOR="#FF0000">G</FONT>raphical <FONT COLOR="#FF0000">O</FONT>perating
<FONT COLOR="#FF0000">S</FONT>ystems
</H1>
<%
QUOT = Chr(34)
strWhere = " WHERE " & Request.Form("criteria")
strDisplay = Request.Form("display")
 %>
<SPAN>Breakdown of operating systems for visitors to our
site<%=strDisplay%>:</SPAN><BR><P>
<%

On Error Resume Next
Set oConn = Server.CreateObject("ADODB.Connection")  'create a connection
Set oRs = Server.CreateObject("ADODB.Recordset")     'create a recordset
oConn.Open strConnLogs              'open connection using value in include file

Function getOSTypeCount(strUADetail)
  strUAWhere = strWhere & strUADetail
  strSQL = "SELECT HitCount=SUM(ItemCount) FROM UserAgentSummary" & strUAWhere
  oRs.Open strSQL, oConn, 3,, 1  'open recordset using adOpenStatic, , adCmdtext
  If Err.Number <> 0 Then
    Response.Write "<FONT FACE=" & QUOT & "Arial" & QUOT & " SIZE=3><B>Sorry, the
database is not accessible at present.</B></FONT>"
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
    Response.End
  End If
  lngResult = 0
  oRs.MoveFirst
  If Not oRs.EOF Then lngResult = oRs("HitCount")
  If lngResult = "" Or IsNull(lngResult) Then lngResult = 0
  oRs.Close
  getOSTypeCount = lngResult
End Function

intTotal = getOSTypeCount("")
intWinNT = getOSTypeCount(" AND (CHARINDEX('Windows NT', ItemText) > 0 OR
CHARINDEX('WinNT', ItemText) > 0 OR CHARINDEX('Windows-NT', ItemText) > 0)")
intWin98 = getOSTypeCount(" AND (CHARINDEX('Windows 98', ItemText) > 0 OR
CHARINDEX('Win98', ItemText) > 0 OR CHARINDEX('Windows-98', ItemText) > 0)")
intWin95 = getOSTypeCount(" AND (CHARINDEX('Windows 95', ItemText) > 0 OR
CHARINDEX('Win95', ItemText) > 0 OR CHARINDEX('Win32', ItemText) > 0)")
intWin3x = getOSTypeCount(" AND (CHARINDEX('Windows 3.', ItemText) > 0 OR
CHARINDEX('Win3.1', ItemText) > 0 OR CHARINDEX('Win16', ItemText) > 0)")
intMacPC = getOSTypeCount(" AND (CHARINDEX('Macintosh', ItemText) > 0 OR
CHARINDEX('Mac_PowerPC', ItemText) > 0)")
intUnix  = getOSTypeCount(" AND (CHARINDEX('(X11', ItemText) > 0 OR CHARINDEX('Unix',
ItemText) > 0) AND CHARINDEX('Linux', ItemText) = 0")
intLinux = getOSTypeCount(" AND CHARINDEX('(X11', ItemText) > 0  AND
CHARINDEX('Linux', ItemText) > 0")
intRISC  = getOSTypeCount(" AND CHARINDEX('Acorn', ItemText) > 0  AND CHARINDEX('RISC
OS', ItemText) > 0")
intOther = intTotal - intWinNT - intWin98 - intWin95 - intWin3x - intMacPC - intUnix -
intLinux - intRISC
If intOther < 10 Then intOther = 10
intTotal = intWinNT + intWin98 + intWin95 + intWin3x + intMacPC + intUnix + intLinux +
intRISC + intOther

Set oConn = Nothing
%>

<%
If blnIE4 Then

'*** create the pie chart ***
iLineCnt = 3
iDegPos = 0
iRectTop = -110
%>
<TABLE STYLE="height:400; width:700" >
<TR><TD>
<OBJECT ID=sgPie STYLE="position:relative; height:230; width:300; top:10; left:10"
  CLASSID="CLSID:369303C2-D7AC-11D0-89D5-00A0C90833E6">
<PARAM NAME="Line0001" VALUE="SetLineColor(0, 0, 0)">
<PARAM NAME="Line0002" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0003" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0004" VALUE="Oval(-130, -90, 200, 200, 0)">

<PARAM NAME="Line0005" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0006" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0007" VALUE="SetFillColor(215, 40, 40)">
<PARAM NAME="Line0008" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(intWinNT / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0009" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intWinNT / intTotal * 360)
iRectTop = iRectTop + 25
```

```
%>
<PARAM NAME="Line0010" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0011" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0012" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0013" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0014" VALUE="SetFillColor(255, 128, 0)">
<PARAM NAME="Line0015" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(intWin98/intTotal*360)) %>, 0)">
<PARAM NAME="Line0016" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intWin98 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0017" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0018" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0019" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0020" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0021" VALUE="SetFillColor(240, 240, 0)">
<PARAM NAME="Line0022" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(intWin95 / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0023" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intWin95 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0024" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0025" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0026" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0027" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0028" VALUE="SetFillColor(30, 200, 47)">
<PARAM NAME="Line0029" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(intWin3x / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0030" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intWin3x / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0031" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0032" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0033" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0034" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0035" VALUE="SetFillColor(30, 203, 203)">
<PARAM NAME="Line0036" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(intMacPC / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0037" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intMacPC / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0038" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0039" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0040" VALUE="Rect(100, <%= CStr(iRectTop) +5 %>, 20, 15, 0)">
<PARAM NAME="Line0041" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0042" VALUE="SetFillColor(80, 80, 218)">
<PARAM NAME="Line0043" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(intUnix / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0044" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intUnix / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0045" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0046" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0047" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0048" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0049" VALUE="SetFillColor(190, 55, 190)">
<PARAM NAME="Line0050" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(CInt(intLinux / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0051" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intLinux / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0052" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0053" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0054" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0055" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0056" VALUE="SetFillColor(128, 128, 0)">
<PARAM NAME="Line0057" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Cint(intRISC / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0058" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intRISC / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0059" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0060" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0061" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0062" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0063" VALUE="SetFillColor(0, 0, 0)">
<PARAM NAME="Line0064" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Cint(intOther / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0065" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
</OBJECT>
</TD><TD WIDTH=5></TD><TD>
<TABLE  CELLSPACING=7>
<TR><TD ALIGN="LEFT">
<B>Windows NT</B>    Count: <B><%= intWinNT %></B>  (<B><%=
FormatPercent(intWinNT / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Windows 98</B>    Count: <B><%= intWin98 %></B>  (<B><%=
FormatPercent(intWin98 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Windows 95</B>    Count: <B><%= intWin95 %></B>  (<B><%=
FormatPercent(intWin95 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Windows 3.x</B>    Count: <B><%= intWin3x %></B>  (<B><%=
FormatPercent(intWin3x / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Mac/PowerPC</B>    Count: <B><%= intMacPC %></B>  (<B><%=
FormatPercent(intMacPC / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Unix (various)</B>    Count: <B><%= intUnix %></B>  (<B><%=
FormatPercent(intUnix / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Linux</B>    Count: <B><%= intLinux %></B>  (<B><%=
FormatPercent(intLinux / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>RISC OS</B>    Count: <B><%= intRISC %></B>  (<B><%=
FormatPercent(intRISC / intTotal, 1) %></B>)
```

Appendix C – Module GraphicOpSys.asp

```
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Other</B>    Count: <B><%= intOther %></B>  (<B><%=
FormatPercent(intOther / intTotal, 1) %></B>)
</TD></TR></TABLE>
</TD></TR></TABLE>

<%
Else
%>

<CENTER><TABLE>
<%
'*** create the table of values ***
Response.Write "<TR><TD>Windows NT    </TD><TD ALIGN=RIGHT><B>    
   " & intWinNT & "</B></TD><TD ALIGN=RIGHT>        " &
FormatPercent(intWinNT / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Windows 98    </TD><TD ALIGN=RIGHT><B>    
   " & intWin98 & "</B></TD><TD ALIGN=RIGHT>        " &
FormatPercent(intWin98 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Windows 95    </TD><TD ALIGN=RIGHT><B>    
   " & intWin95 & "</B></TD><TD ALIGN=RIGHT>        " &
FormatPercent(intWin95 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Windows 3.x    </TD><TD ALIGN=RIGHT><B>  
     " & intWin3x & "</B></TD><TD ALIGN=RIGHT>      
 " & FormatPercent(intWin3x / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Mac/PowerPC    </TD><TD ALIGN=RIGHT><B>  
     "  & intMacPC & "</B></TD><TD ALIGN=RIGHT>      
 "  & FormatPercent(intMacPC / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Unix (various)    </TD><TD ALIGN=RIGHT><B>  
     "  & intUnix & "</B></TD><TD ALIGN=RIGHT>      
 "  & FormatPercent(intUnix / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Linux    </TD><TD ALIGN=RIGHT><B>    
   "  & intLinux & "</B></TD><TD ALIGN=RIGHT>        "
& FormatPercent(intLinux / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>RISC OS    </TD><TD ALIGN=RIGHT><B>    
   "  & intRISC & "</B></TD><TD ALIGN=RIGHT>        "  &
FormatPercent(intRISC / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Other User Agents    </TD><TD ALIGN=RIGHT><B>  
     " & intOther & "</B></TD><TD ALIGN=RIGHT>      
 " & FormatPercent(intOther / intTotal, 1) & "</TD></TR>" & vbCrlf
%>
</TABLE></CENTER>

<%
End If
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module GraphicOpSys.asp

Module Name – GraphicUaType.asp

```
<%@ Language=VBScript %>
<% Server.ScriptTimeOut = 600 %>

<%
blnIE4 = False
strUA = Request.ServerVariables("HTTP_USER_AGENT")
intMSIE = Instr(strUA, "MSIE ") + 5
If intMSIE > 5 Then
  If CInt(Mid(strUA, intMSIE, Instr(intMSIE, strUA, ".") - intMSIE)) >= 4 Then
    blnIE4 = True
  End If
End If
%>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Results of your traffic query</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="TRAFFICUAT"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
   if Not IsAdmin() then
      Response.Write("<B>Sorry, this function is not available </B>")
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
    Response.End
   end if
%>
<H1 ALIGN="center">
<FONT COLOR="#FF0000">G</FONT>raphical <FONT COLOR="#FF0000">B</FONT>rowser
<FONT COLOR="#FF0000">T</FONT>ypes
</H1>
<%
QUOT = Chr(34)
strWhere = " WHERE " & Request.Form("criteria")
strDisplay = Request.Form("display")
 %>
<SPAN>Breakdown of browser types visiting our server <%=strDisplay%>:</SPAN><BR><P>
<%

On Error Resume Next
Set oConn = Server.CreateObject("ADODB.Connection")  'create a connection
Set oRs = Server.CreateObject("ADODB.Recordset")     'create a recordset
oConn.Open strConnLogs              'open connection using value in include file

Function getUATypeCount(strUADetailIn, intUADetailMaxPosn, strUADetailNotIn1,
strUADetailNotIn2)
  strUAWhere = strWhere
  If Len(strUADetailIn) Then strUAWhere = strUAWhere & " AND CHARINDEX('" &
strUADetailIn & "', ItemText) > 0"
  If intUADetailMaxPosn > 0 Then strUAWhere = strUAWhere & " AND CHARINDEX('" &
strUADetailIn & "', ItemText) < " & CStr(intUADetailMaxPosn + 1)
  If Len(strUADetailNotIn1) Then strUAWhere = strUAWhere & " AND CHARINDEX('" &
strUADetailNotIn1 & "', ItemText) = 0"
  If Len(strUADetailNotIn2) Then strUAWhere = strUAWhere & " AND CHARINDEX('" &
strUADetailNotIn2 & "', ItemText) = 0"
  strSQL = "SELECT HitCount=SUM(ItemCount) FROM UserAgentSummary" & strUAWhere
  oRs.Open strSQL, oConn, 3,, 1  'open recordset using adOpenStatic, , adCmdtext
  If Err.Number <> 0 Then
    Response.Write "<B>The database is not accessible at present</B>"
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
    Response.End
  End If
  lngResult = 0
  oRs.MoveFirst
  If Not oRs.EOF Then lngResult = oRs("HitCount")
  If lngResult = "" Or IsNull(lngResult) Then lngResult = 0
  oRs.Close
  getUATypeCount = lngResult
End Function

intTotal = getUATypeCount(" ", 0, "", "")
intIE3 = getUATypeCount("(compatible; MSIE 3.", 40, "", "")
intIE4 = getUATypeCount("(compatible; MSIE 4.", 40, "", "")
intIE5 = getUATypeCount("(compatible; MSIE 5.", 40, "", "")
intNav2 = getUATypeCount("Mozilla/2.", 1, "MSIE", "Opera")
intNav3 = getUATypeCount("Mozilla/3.", 1, "MSIE", "Opera")
intNav4 = getUATypeCount("Mozilla/4.", 1, "MSIE", "Opera")
intNav5 = getUATypeCount("Mozilla/5.", 1, "MSIE", "Opera")
intOpera = getUATypeCount("Opera", 0, "", "")

intOther = intTotal - intIE3 - intIE4 - intIE5 - intNav2 - intNav3 - intNav4 - intNav5
- intOpera
If intOther < 1 Then intOther = 1
intTotal = intIE3 + intIE4 + intIE5 + intNav2 + intNav3 + intNav4 + intNav5 + intOpera
+ intOther
Set oConn = Nothing
%>


<%
If blnIE4 Then

'*** create the pie chart ***
iLineCnt = 3
iDegPos = 0
iRectTop = -110
%>
<TABLE STYLE="height:400; width:700" >
<TR><TD>
<OBJECT ID=sgPie STYLE="position:relative; height:230; width:300; top:10; left:10"
  CLASSID="CLSID:369303C2-D7AC-11D0-89D5-00A0C90833E6">
<PARAM NAME="Line0001" VALUE="SetLineColor(0, 0, 0)">
<PARAM NAME="Line0002" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0003" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0004" VALUE="Oval(-130, -90, 200, 200, 0)">

<PARAM NAME="Line0005" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0006" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0007" VALUE="SetFillColor(128, 128, 0)">
<PARAM NAME="Line0008" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intNav2 / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0009" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
```

Appendix C – Module GraphicUaType.asp

```
iDegPos = iDegPos + CInt(intNav2 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0010" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0011" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0012" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0013" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0014" VALUE="SetFillColor(190, 55, 190)">
<PARAM NAME="Line0015" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intNav3/intTotal*360)) %>, 0)">
<PARAM NAME="Line0016" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intNav3 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0017" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0018" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0019" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0020" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0021" VALUE="SetFillColor(80, 80, 218)">
<PARAM NAME="Line0022" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intNav4 / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0023" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intNav4 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0024" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0025" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0026" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0027" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0028" VALUE="SetFillColor(30, 203, 203)">
<PARAM NAME="Line0029" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intNav5 / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0030" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intNav5 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0031" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0032" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0033" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0034" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0035" VALUE="SetFillColor(30, 200, 47)">
<PARAM NAME="Line0036" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intIE3 / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0037" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intIE3 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0038" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0039" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0040" VALUE="Rect(100, <%= CStr(iRectTop) +5 %>, 20, 15, 0)">
<PARAM NAME="Line0041" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0042" VALUE="SetFillColor(240, 240, 0)">
<PARAM NAME="Line0043" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intIE4 / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0044" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intIE4 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0045" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0046" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0047" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0048" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0049" VALUE="SetFillColor(255, 128, 0)">
<PARAM NAME="Line0050" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intIE5 / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0051" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intIE5 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0052" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0053" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0054" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0055" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0056" VALUE="SetFillColor(215, 40, 40)">
<PARAM NAME="Line0057" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intOpera / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0058" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
<%
iDegPos = iDegPos + CInt(intIE5 / intTotal * 360)
iRectTop = iRectTop + 25
%>
<PARAM NAME="Line0059" VALUE="SetFillColor(192, 192, 192">
<PARAM NAME="Line0060" VALUE="SetLineStyle(0)">
<PARAM NAME="Line0061" VALUE="Rect(100, <%= CStr(iRectTop) + 5 %>, 20, 15, 0)">
<PARAM NAME="Line0062" VALUE="SetLineStyle(1)">
<PARAM NAME="Line0063" VALUE="SetFillColor(0, 0, 0)">
<PARAM NAME="Line0064" VALUE="Pie(-140, -100, 200, 200, <%= CStr(iDegPos) %>, <%=
CStr(Round(intOther / intTotal * 360)) %>, 0)">
<PARAM NAME="Line0065" VALUE="Rect(95, <%= CStr(iRectTop) %>, 20, 15, 0)">
</OBJECT>
</TD><TD WIDTH=5></TD><TD>
<TABLE  CELLSPACING=7>
<TR><TD ALIGN="LEFT">
<B>Netscape Navigator 2.x</B>    Count: <B><%= intNav2 %></B>  (<B><%=
FormatPercent(intNav2 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Netscape Navigator 3.x</B>    Count: <B><%= intNav3 %></B>  (<B><%=
FormatPercent(intNav3 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Netscape Navigator 4.x</B>    Count: <B><%= intNav4 %></B>  (<B><%=
FormatPercent(intNav4 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Netscape Navigator 5.x</B>    Count: <B><%= intNav5 %></B>  (<B><%=
FormatPercent(intNav5 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Internet Explorer 3.x</B>    Count: <B><%= intIE3 %></B>  (<B><%=
FormatPercent(intIE3 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Internet Explorer 4.x</B>    Count: <B><%= intIE4 %></B>  (<B><%=
FormatPercent(intIE4 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Internet Explorer 5.x</B>    Count: <B><%= intIE5 %></B>  (<B><%=
FormatPercent(intIE5 / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
```

Appendix C – Module GraphicUaType.asp

```
<B>Opera (all versions)</B>    Count: <B><%= intOpera %></B>  (<B><%=
FormatPercent(intOpera / intTotal, 1) %></B>)
</TD></TR>
<TR><TD ALIGN="LEFT">
<B>Other User Agents</B>    Count: <B><%= intOther %></B>  (<B><%=
FormatPercent(intOther / intTotal, 1) %></B>)
</TD></TR></TABLE>
</TD></TR></TABLE>

<%
Else
%>

<CENTER><TABLE>
<%
'*** create the table of values ***
Response.Write "<TR><TD>Netscape Navigator 2.x    </TD><TD ALIGN=RIGHT><B>
       " & intNav2 & "</B></TD><TD ALIGN=RIGHT>    
   " & FormatPercent(intNav2 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Netscape Navigator 3.x    </TD><TD ALIGN=RIGHT><B>
       " & intNav3 & "</B></TD><TD ALIGN=RIGHT>    
   " & FormatPercent(intNav3 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Netscape Navigator 4.x    </TD><TD ALIGN=RIGHT><B>
       " & intNav4 & "</B></TD><TD ALIGN=RIGHT>    
   " & FormatPercent(intNav4 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Netscape Navigator 5.x    </TD><TD ALIGN=RIGHT><B>
       " & intNav5 & "</B></TD><TD ALIGN=RIGHT>    
   " & FormatPercent(intNav5 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Internet Explorer 3.x    </TD><TD ALIGN=RIGHT><B>
       "  & intIE3 & "</B></TD><TD ALIGN=RIGHT>    
   "  & FormatPercent(intIE3 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Internet Explorer 4.x    </TD><TD ALIGN=RIGHT><B>
       "  & intIE4 & "</B></TD><TD ALIGN=RIGHT>    
   "  & FormatPercent(intIE4 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Internet Explorer 5.x    </TD><TD ALIGN=RIGHT><B>
       "  & intIE5 & "</B></TD><TD ALIGN=RIGHT>    
   "  & FormatPercent(intIE5 / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Opera (all versions)    </TD><TD ALIGN=RIGHT><B>
       "  & intOpera & "</B></TD><TD ALIGN=RIGHT>    
   "  & FormatPercent(intOpera / intTotal, 1) & "</TD></TR>" & vbCrlf
Response.Write "<TR><TD>Other User Agents    </TD><TD ALIGN=RIGHT><B>  
     " & intOther & "</B></TD><TD ALIGN=RIGHT>      
 " & FormatPercent(intOther / intTotal, 1) & "</TD></TR>" & vbCrlf
%>
</TABLE></CENTER>

<%
End If
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module GraphicUaType.asp

Module Name – Login.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<%
  Dim PageName
  PageName="LOGON"
%>
<HEAD>
<TITLE>Logon</TITLE>
</HEAD>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<Hl ALIGN="center"><FONT
COLOR="#FF0000">L</FONT>ogin
</Hl>

<%
Dim strEMailAddress
Dim strPassword
Dim intPassCount
Dim strResult
sub TermMess( theMess)
%>
<%=theMess%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
   Response.end
end sub

if Session("Locked") = true then
  TermMess "Sorry you have been locked out for the duration of the current session."
end if

intPassCount=Request.Form.Item("pc")
if Len(intPassCount) < 1 Then ' first time
  intPassCount=3
end if


strEmailAddress = Request.Form.Item("v1")
strPassword = Request.Form.Item("v2")
On Error Resume Next
if Len(strEmailAddress) > 0 then ' decrement pass count for following test
   intPassCount = IntPassCount-1
   if (intPassCount < 1 or intPassCount > 2) then 'lock user out
     Session("Locked") = true
     TermMess "You have exceeded the number of logon attempts allowed and are now
locked out for the current session."
   End If 'lockuser out
' check to see if user is valid

   'create and open a connection to the database
   Err.Clear ()
   Set oCon = Server.CreateObject("ADODB.Connection")
   if Err.number > 0 then TermMess "Unable to connect to database(1)"
   Err.Clear ()
   oCon.Open strConnection
   if Err.number > 0 then TermMess "Unable to connect to database(2)"

   'create a command and set the properties
   Err.Clear ()
   Set oCmd = Server.CreateObject("ADODB.Command")
   if Err.number > 0 then TermMess "Unable to connect to database(3)"
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4            'stored procedure
   oCmd.CommandText = "ValidateUser"   'procedure name

   'execute the command, supplying the parameters, and get the result
   Err.Clear ()
   Set oRs = oCmd.Execute(lngRecsAffected, Array(strEmailAddress, strPassword))
   if Err.number > 0 then TermMess "Unable to execute command on database"

   if not ORs.Eof then
'display the result
     Session("UserLogon")=strEmailAddress
     Session("UsePass")=strPassWord 'Kept as parameter for procedures
GetUserDetails,and ModUsers
     Session("UseriD")=Ors.Fields("User_ID")
     Session("UserRole")=oRs.Fields("UserRole")
     TermMess "You are logged on as " & Session("UserLogon") & ". If you would like to
modify your details <A HREF=""modLogon.asp"">please click here</A>. "
   End if
%>

<P>The login details you have entered cannot be verified.
You have <%=intPassCount%> attempts left allowed so please try again. If you have
forgotten your password, <A HREF="LOGONPASS.asp?UI=<%=strEmailAddress%>">please click
here</A>
and it will be emailed to you. subscribe .</P>
<%
end if
%>
<P>Annual subscriptions to JIPAM are currently free. To download full text articles
as PDF files, you must <A HREF="Subscribe.asp">subscribe</A> to JIPAM.</P>
<P>If you are currently a JIPAM subscriber and wish to access the full text articles,
enter your login details below.</P><BR>
<BR>
<%
   'display text boxes to collect details
%>

   <form action="login.Asp" method="POST">
   <center>
   <table>
   <tr>
    <td>Email Address:</td>
    <td><input type="text" size="48" name="v1" value="<%=strEMailAddress%>"
MAXLENGTH="128"></td>
   </tr>
   <tr>
    <td>Password:</td>
    <td><input type="password" size="48" name="v2" MAXLENGTH="32"></td>
   </tr>
   <tr>
    <td></td>
    <td align="right"><input type="submit" value="Login"></td>
   </tr>
   </table>
   <input TYPE="HIDDEN" NAME="pc" Value="<%=intPassCount%>">
   </center>
   </form>
```

Appendix C – Module Login.asp

```
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module Login.asp

Module Name – LogonPass.asp

```
<%@ Language=VBScript %>
<!-- #include virtual="/jipam/common/common.inc" -->
<% ' this process emails a user password
  Dim PageName
  Dim strUserName
  PageName="LOGON"
%>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<HEAD>
<TITLE>Down Map</TITLE>
</HEAD>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
  strUserName = Request.QueryString("UI")
  if len(strUserName) < 1  then
%>
<I>Your Email address in valid.</I>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
  Response.End
  end if

   'create and open a connection to the database
   Set oCon = Server.CreateObject("ADODB.Connection")
   oCon.Open strConnection
  'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   oCmd.ActiveConnection = oCon
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = "SendUserPassword"   'procedure name
   lngRecsAffected=0
   arrParams=Array(strUserName)
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute(lngRecsAffected,arrParams )
%>
<I>Your pássword has been sent to your email address.</I>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Module Name – MainHeader.inc

```
<BODY BACKGROUND="/jipam/images/bgbeige2.gif">
<TABLE WIDTH="100%">
<TBODY>
<TR>
<TD ALIGN="LEFT"> <TABLE WIDTH="100%" ALIGN="LEFT">
<TR>
<TD WIDTH="300" ALIGN="LEFT">
<!--before masthead-->
<MAP NAME="FPMap0">
<AREA COORDS="0,9,129,147" HREF="http://tm/jipam/default.asp" SHAPE="RECT">
<AREA COORDS="21,157,105,163" HREF="http://www.issn.org/" SHAPE="RECT">
</MAP>
<IMG ALT="JIPAM Logo" BORDER="0" HEIGHT="164" SRC="/jipam/images/JIPAMlogo.gif"
 USEMAP="#FPMap0" WIDTH="130" ALIGN="LEFT"></TD>
<TD ALIGN="LEFT"> <MAP NAME="FPMap1">
<AREA COORDS="380,130,558,150" HREF="http://rgmia.vu.edu.au/SSDragomirWeb.html"
 SHAPE="RECT">
</MAP>
<IMG ALT="JIPAM Masthead" BORDER="0" HEIGHT="168"
 SRC="/jipam/images/JIPAMmast.gif" USEMAP="#FPMap1" WIDTH="559" ALIGN="LEFT">
 <!--after masthead-->
</TD>
</TR>
</TABLE>
</TD>
</TR>
<TR>
<TD> <TABLE WIDTH="100%">
<TBODY>
<TR>
<TD VALIGN="top" WIDTH="130"> <CENTER>
<!--before side menu-->
```

MODULE NAME – MANAGEMENT.ASP

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Management</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="MANAGEMENT"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<table border="0" width="100%" cellspacing="0" cellpadding="0">
  <tr>
    <td>
      <h1 align="center"><font color="#FF0000"><font size="6"><a
name="board"></a></font>M</font>anagerial
      <font color="#FF0000">B</font>oard</hl>
      <p><em>JIPAM</em> is run by a small, professional and
      dedicated team   The current membership of the Managerial Board
      is:</p>
      <blockquote>
        <blockquote>
          <blockquote>
 <table border="1" bordercolor="#000000" bgcolor="#FFFFCC" cellpadding="10">
            <tr>
              <td nowrap>
                <ul>
                  <li>Neil S. Barnett - Editor</li>
                  <li>Pietro Cerone - Editor</li>
                  <li>Sever S. Dragomir - Editor-in-Chief</li>
                  <li>John Roumeliotis - Managing Editor</li>
                  <li><a href="mailto:Anthony.Sofo@vu.edu.au">Anthony Sofo</a> -
Chair</li>
                </ul>
              </td>
            </tr>
          </table>
          </blockquote>
        </blockquote>
      </blockquote>
      <p>
      The board is responsible for the direction of the journal including
      setting the policy framework under which the journal operates.</p>
      <p>The management team is:</p>
          <blockquote>
        <blockquote>
        <table border="1" bordercolor="#000000" cellpadding="10" bgcolor="#FFFFCC">
          <tr>
            <td width="50%" valign="top" nowrap>
              <ul>
              <li> Neil S. Barnett
                <li>Pietro Cerone</li>
                <li>Sever S. Dragomir</li>
                <li>Brendan Hack</li>
                <li>George Hanna</li>
                <li>Jerry Koliha</li>
                <li>Alasdair McAndrew</li>
                              </ul>
            </td>
            <td width="50%" valign="top" nowrap>
              <ul>
                <li>Anna Pang</li>
                <li>John Roumeliotis</li>
                <li>Anthony Sofo</li>
                <li>Audrey Tam</li>
                <li>Peter Vouzas</li>
                <li>Stephen Young</li>
              </ul>
            </td>
          </tr>
        </table>
        </blockquote>
      </blockquote>
      <p>Listed below are the sub-committees that have responsibility for
      certain areas of operation.  An asterisk indicates a chair of the
      sub-committee.</p>
      <p>If you think our operation can be improved in any way, please contact
      the appropriate person.</p>
      <p>Click here to see <a href="managementphotos.asp">photographs</a> of
      the <em>JIPAM</em> Managerial Board</p>
      <blockquote>
        <table border="1" width="80%" bgcolor="#FFFFCC">
          <tr>
            <td width="80%" bordercolorlight="#000000">
              <p align="center"><strong><a
name="admin"></a>Administration</strong></p>
              <ul>
                <li>(*) Anna Pang <a href="mailto:anna@cabsav.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
                <li><a href="http://sci.vu.edu.au/~vouzas" target="_top">Peter
                  Vouzas</a> <a href="mailto:vouzas@matilda.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
              </ul>
              <p> </td>
          </tr>
          <tr>
            <td width="80%" bordercolorlight="#000000">
              <p align="center"><strong><a name="article"></a>Article
              Preparation, Format and Design</strong></p>
              <ul>
                <li><a href="http://sci.vu.edu.au/~amca">Alasdair McAndrew</a> <a
href="mailto:amca@matilda.vu.edu.au"><img src="images/mailto.gif" alt="mailto.gif (96
bytes)" border="0" width="14" height="10"></a></li>
                <li>Anna Pang <a href="mailto:anna@cabsav.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
                <li>(*) <a href="http://dingo.vu.edu.au/~johnr">John
Roumeliotis</a>  <a href="mailto:John.Roumeliotis@vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
              </ul>
              <p> </td>
          </tr>
          <tr>
            <td>
              <p align="center"><strong><a name="publicity"></a>Publicity</strong></p>
              <ul>
```

APPENDIX C – MODULE MANAGEMENT.ASP

```
                <li><a href="http://sci.vu.edu.au/staff/peterc.html"
target="_top">Pietro
                  Cerone</a> <a href="mailto:pc@matilda.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
                <li>(*) <a href="http://rgmia.vu.edu.au/dragomirweb.html"
target="_top">Sever
                  Dragomir</a> <a href="mailto:sever@matilda.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
                <li><a href="http://sci.vu.edu.au/~georgey" target="_top">George
                  Hanna</a> <a href="mailto:georgey@matilda.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
                <li><a
href="http://macserver.maths.mu.oz.au/homepage.taf?function=list&amp;Surname=KOLIHA&am
p;FirstName=Jerry" target="_top">Jerry
                  Koliha</a> <a href="mailto:J.Koliha@ms.unimelb.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
                <li><a href="http://sci.vu.edu.au/staff/anthonys.html"
target="_top">Anthony
                  Sofo</a> <a href="mailto:sofo@matilda.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
              </ul>
              <p> </td>
          </tr>
          <tr>
            <td>
              <p align="center"><strong><a name="web"></a>Web Design,
              Maintenance</strong></p>
              <ul>
                <li>Brendan Hack <a href="mailto:bendy@cabsav.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
                <li>(*) <a href="http://dingo.vu.edu.au/~johnr">John Roumeliotis</a>
<a href="mailto:John.Roumeliotis@vu.edu.au"><img src="images/mailto.gif"
alt="mailto.gif (96 bytes)" border="0" width="14" height="10"></a></li>
                <li>Peter Vouzas</li>
                <li><a href="http://sci.vu.edu.au/~amt" target="_top">Audrey Tam</a>
                  <a href="mailto:amt@matilda.vu.edu.au"><img src="images/mailto.gif"
alt="mailto.gif (96 bytes)" border="0" width="14" height="10"></a></li>
                <li><a href="http://sci.vu.edu.au/staff/stepheny.html"
target="_top">Stephen
                  Young</a> <a href="mailto:stephen@matilda.vu.edu.au"><img
src="images/mailto.gif" alt="mailto.gif (96 bytes)" border="0" width="14"
height="10"></a></li>
              </ul>
              <p> </td>
          </tr>
        </table>
      </blockquote>
    </td>
  </tr>
</table>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module Management.asp

Module Name – ManagementPhotos.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Management Photos</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="MANAGEMENTP"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<table border="0" width="100%" cellspacing="0" cellpadding="0">
  <tr>
    <td>
      <h1 align="center"><font color="#FF0000">M</font>anagement</h1>
      <table border="1" width="100%" cellpadding="5">
        <tr>
          <td width="70%">
            <p align="center"><img border="0" src="images/bneil.jpg" width="170"
height="202"></td>
          <td width="30%">
            <p align="center"><big>Neil Barnett</big><br>
            (Editor, Managerial Board)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/cpeter.jpg" width="159"
height="212"></td>
          <td width="30%">
            <p align="center"><big>Pietro Cerone</big><br>
            (Editor, Managerial Board, Publicity)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/sever.jpg" alt="sever.jpg" width="150"
height="143"></td>
          <td width="30%">
            <p align="center"><big>Sever Dragomir</big><br>
            (Editor-in-Chief, Managerial Board, Publicity)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/Dc0027I.JPG" width="101" height="152"
alt="Dc0027I.JPG "></td>
          <td width="30%">
            <p align="center"><big>Brendan Hack</big><br>
            (Web Design)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/george.jpg" width="160"
height="214"></td>
          <td width="30%">
            <p align="center"><big>George Hanna<br>
            </big>(Publicity)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/jerry.jpg" alt="jerry.jpg (9158 bytes)"
width="128" height="192"></td>
          <td width="30%">
            <p align="center"><big>Jerry Koliha<br>
            </big>(Publicity)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/Dc0065I.JPG" alt="Dc0065I.JPG (5915
bytes)" width="90" height="135"></td>
          <td width="30%">
            <p align="center"><big>Alasdair McAndrew<br>
            </big>(Article Design)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/john.jpg" alt="john.jpg (3130 bytes)"
width="150" height="162"></td>
          <td width="30%">
            <p align="center"><big>John Roumeliotis<br>
            </big>(Managing Editor, Managerial Board, Article Design, Web
            Design) </td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/Dc0029I.JPG" width="108" height="162"
alt="Dc0029I.JPG (5916 bytes)"></td>
          <td width="30%">
            <p align="center"><big>Anthony Sofo<br>
            </big>(Chair of Managerial Board, Publicity)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/ATam.gif" width="105" height="133"
alt="me.gif (12234 bytes)"></td>
          <td width="30%">
            <p align="center"><big>Audrey Tam<br>
            </big>(Web Design)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/peterv.gif" width="80" height="100"
alt="peterv.gif (4384 bytes)"></td>
          <td width="30%">
            <p align="center"><big>Peter Vouzas<br>
            </big>(Web Design)</td>
        </tr>
        <tr>
          <td width="70%">
            <p align="center"><img src="images/ystephen.jpg" width="174" height="232"
alt="ystephen.jpg (74957 bytes)"></td>
          <td width="30%">
            <p align="center"><big>Stephen Young<br>
            </big>(Web Design)</td>
        </tr>
      </table>
    </td>
  </tr>
</table>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module ManagementPhotos.asp

MODULE NAME – MENU.INC

```
<!--before side menu-->
<TABLE WIDTH="100%">
<TBODY>
<TR>
<MAP NAME="FPMain"><AREA COORDS="0,0,125,35" HREF="/jipam/default.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="HOME" BORDER="0" HEIGHT="35" WIDTH="125"
<%   if PageName="MAIN" then %>
SRC="/jipam/images/Home-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/Home-up.gif" USEMAP="#FPMain" > </TD>
<%  end if  %>
</TR>
<TR>
<MAP NAME="FPAbout"><AREA COORDS="0,0,125,35" HREF="/jipam/AimsScope.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="ABOUT JIPAM" BORDER="0" HEIGHT="35"
WIDTH="125"
<%   if PageName="SCOPE" then %>
SRC="/jipam/images/AboutJIPAM-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/AboutJIPAM-up.gif" USEMAP="#FPAbout" > </TD>
<%  end if  %>
</TR>
<TR>
<MAP NAME="FPCopyright"><AREA COORDS="0,0,125,35" HREF="/jipam/copyright.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="COPYRIGHT" BORDER="0" HEIGHT="35"
WIDTH="125"
<%   if PageName="COPYRIGHT" then %>
SRC="/jipam/images/Copyright-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/Copyright-up.gif" USEMAP="#FPCopyright" > </TD>
<%  end if  %>
</TR>
<TR>
<MAP NAME="FPEditors"><AREA COORDS="0,0,125,35" HREF="/jipam/Editors.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="EDITORS" BORDER="0" HEIGHT="35" WIDTH="125"
<%   if PageName="EDITORS" then %>
SRC="/jipam/images/Editors-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/Editors-up.gif" USEMAP="#FPEditors" > </TD>
<%  end if  %>
</TR>
<TR>
<MAP NAME="FPWhatsNew"><AREA COORDS="0,0,125,35" HREF="/jipam/WhatsNew.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="WHATS NEW" BORDER="0" HEIGHT="35"
WIDTH="125"
SRC="/jipam/images/WhatsNew-up.gif"
> </TD>
</TR>
<TR>
<TD WIDTH="100%"> </TD>
</TR>
<TR>
<MAP NAME="FPCurrentIss"><AREA COORDS="0,0,125,35" HREF="/jipam/DispVolume.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="CURRENT ISSUE" BORDER="0" HEIGHT="35"
WIDTH="125"
<%   if PageName="DISPLAYVOL" then %>
SRC="/jipam/images/CurrentIssue-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/CurrentIssue-up.gif" USEMAP="#FPCurrentIss" > </TD>
<%  end if  %>
</TR>
<TR>
<MAP NAME="FPNewPapers"><AREA COORDS="0,0,125,35" HREF="/jipam/Accepted.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="NEW PAPERS" BORDER="0" HEIGHT="35"
WIDTH="125"
<%   if PageName="ACCEPTED" then %>
SRC="/jipam/images/NewPapers-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/NewPapers-up.gif" USEMAP="#FPNewPapers" > </TD>
<%  end if  %>
</TR>
<TR>
<MAP NAME="FPSearch"><AREA COORDS="0,0,125,35" HREF="/jipam/Search.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="SEARCH" BORDER="0" HEIGHT="35" WIDTH="125"
<%   if PageName="SEARCH" then %>
SRC="/jipam/images/Search-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/Search-up.gif" USEMAP="#FPSearch" > </TD>
<%  end if  %>
</TR>
<% if not IsLoggedOn() then %>
<TR>
<MAP NAME="FPLogin"><AREA COORDS="0,0,125,35" HREF="/jipam/login.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="LOGIN" BORDER="0" HEIGHT="35" WIDTH="125"
<%   if PageName="LOGON" then %>
SRC="/jipam/images/Login-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/Login-up.gif" USEMAP="#FPLogin" > </TD>
<%  end if  %>
</TR>
<% end if %>
<TR>
<MAP NAME="FPAuthorInf"><AREA COORDS="0,0,125,35" HREF="/jipam/Authors.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="AUTHOR INFO" BORDER="0" HEIGHT="35"
WIDTH="125"
<%   if PageName="AUTHORS" then %>
SRC="/jipam/images/AuthorInfo-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/AuthorInfo-up.gif" USEMAP="#FPAuthorInf" > </TD>
<%  end if  %>
</TR>
<TR>
<TD WIDTH="100%"> </TD>
</TR>
<TR>
<MAP NAME="FPPrize"><AREA COORDS="0,0,125,35" HREF="/jipam/Authors.asp#Prize"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="PRIZE" BORDER="0" HEIGHT="35" WIDTH="125"
<%   if PageName="AUTHORS" then %>
SRC="/jipam/images/Prize-down.gif"> </TD>
<%   else  %>
SRC="/jipam/images/Prize-up.gif" USEMAP="#FPPrize" > </TD>
<%  end if  %>
```

APPENDIX C – MODULE MENU.INC

```
</TR>
<TR>
<TD WIDTH="100%"> </TD>
</TR>
<TR>
<MAP NAME="FPFeedback"><AREA COORDS="0,0,125,35" HREF="/jipam/Feedback.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="FEEDBACK" BORDER="0" HEIGHT="35" WIDTH="125"
<%   if PageName="FEEDBACK" then %>
SRC="/jipam/images/Feedback-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/Feedback-up.gif" USEMAP="#FPFeedback" > </TD>
<%  end if  %>
</TR>
<TR>
<MAP NAME="FPContact"><AREA COORDS="0,0,125,35" HREF="/jipam/ContactUs.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="CONTACT US" BORDER="0" HEIGHT="35"
WIDTH="125"
<%   if PageName="CONTACT" then %>
SRC="/jipam/images/ContactUs-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/ContactUs-up.gif" USEMAP="#FPContact" > </TD>
<%  end if  %>
</TR>
<%  if IsAdmin() then %>
<TR>
<MAP NAME="FPReports"><AREA COORDS="0,0,125,35" HREF="/jipam/TrafficReports.asp"
SHAPE="RECT"></MAP>
<TD WIDTH="100%"><IMG ALIGN="center" ALT="Visitor Reports" BORDER="0" HEIGHT="35"
WIDTH="125"
<%   if PageName="TRAFFIC" then %>
SRC="/jipam/images/Reports-down.gif"  > </TD>
<%   else  %>
SRC="/jipam/images/reports-up.gif" USEMAP="#FPReports" > </TD>
<%  end if  %>
</TR>

<% end if %>
</TBODY>
</TABLE>
<!--after side menu-->
</CENTER>
</TD>
<TD VALIGN="top" WIDTH="12"></TD>
<TD> <TABLE WIDTH="100%">
<TBODY>
<TR>
<TD>
<!--before contents-->
```

Appendix C – Module Menu.Inc

Module Name – ModLogon.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<!-- #include virtual="/jipam/common/error.inc" -->
<HEAD>
<TITLE>Site Map</TITLE>
</HEAD>
<BODY >
<%
Sub TerminateWithMessage(Mess)
   if len(Mess) > 0 then Response.Write(Mess)
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
  Response.End
end Sub

  Dim PageName
  PageName="MODLOGON"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
   if Session("Locked") or len(Session("UserLogon")) < 1 or Session("UseriD") < 101
then ' not called by logon.asp
     TerminateWithMessage "<H2>Sorry this facility is not
available.</H2><P></P><P></P>"
     Response.End
    end if
    ' we should now check to see if we are updating the data rather than retrieving it
  On Error Resume Next
  Dim strEMailAddress
  Dim strPassWord
  Dim strFirstName
  Dim strOtherInits
  Dim strSurname
  Dim strWebAddress
  Dim strAffiliation
  Dim strDisplayName 'NB this field displays only on modlogon.asp
  Dim intResult
  intResult=0
  'create and open a connection to the database
  Err.Clear()
  Set oCon = Server.CreateObject("ADODB.Connection")
  if Err.number > 0 then TerminateWithMessage "Cannot connect to Database(1)"
  oCon.Open strConnection
  if Err.number > 0 then TerminateWithMessage "Cannot connect to Database(2)"
 'create a command and set the properties
  Set oCmd = Server.CreateObject("ADODB.Command")
  if Err.number > 0 then TerminateWithMessage "Cannot connect to Database(3)"
  oCmd.ActiveConnection = oCon
  oCmd.CommandType = 4           'stored procedure
  if Request.Form.Count > 1 then 'this is the second pass
    On Error Resume Next
    strEMailAddress = trim(Request.Form.Item("E1"))
    strPassWord = trim(Request.Form.Item("E2"))
    strFirstName = trim(Request.Form.Item("E4"))
    strOtherInits = trim(Request.Form.Item("E5"))
    strSurname = trim(Request.Form.Item("E6"))
    strWebAddress = trim(Request.Form.Item("E7"))
    strAffiliation = trim(Request.Form.Item("E8"))
    if Session("UserRole") > 0 then
       strDisplayName=trim(Request.Form.Item("E9"))
    else
       strDisplayName=""
    end if
    oCmd.CommandText = "ModUserLong"   'procedure name
    lngRecsAffected=0
    Err.Clear ()
    arrParams=Array(Session("UserId"),Session("UsePass"),strEMailAddress, _

strPassWord,strFirstName,strOtherInits,strSurname,strWebAddress,strAffiliation,strDisp
layName)
    if Err.number > 0 then TerminateWithMessage "Cannot create array"
   'execute the command, supplying the parameters, and get the result
   Err.Clear()
   Set oRs = oCmd.Execute(lngRecsAffected,arrParams )
   if Err.number > 0 then TerminateWithMessage "Cannot execute command on Database(1)"

   if Ors.Eof then ' failure
%>
<H2>Sorry, there has been an internal error and your details could not be
changed.</H2>
<P></P><P>Please contact the JIPAM <A
HREF="mailto:<%=Application.Contents.Item("EmailAdmin")%>">administrator</A></P><P></P
><HR>
<%
      Mess=""
      TerminateWithMessage Mess
      Response.End
    else
     TerminateWithMessage "<H2>Your details have been changed.</H2><P><HR>"
     Response.End
    end if
    TermWithMessage "A problem occurred updating your new  details<P><HR>"
    Response.End

 end if 'Request.Form.Count
 'this is first entry into form

  oCmd.CommandText = "GetUserDetails"   'procedure name
  lngRecsAffected=0
  'execute the command, supplying the parameters, and get the result
  Set oRs = oCmd.Execute(lngRecsAffected,array(Session("UserId"),Session("UsePass")) )
  if Err.number > 0 then TerminateWithMessage "Cannot execute command on Database(2)"
  if oRs.Eof then ' should only happen rarely if at all
   ' because it means that the record in the database has been changed since user
logged on
%>
<H2>Sorry, there has been an internal error.</H2>
<P></P><P>Please contact the JIPAM <A
HREF="mailto:<%=Application.Contents.Item("EmailAdmin")%>">administrator</A></P><P></P
><HR>
<%
      Mess=""
      TerminateWithMessage Mess
      Response.End
  end if
  ' get users current details
  strEMailAddress=trim(oRs.Fields("EmailAddress"))
  strPassWord=trim(Session("UsePass"))
  strFirstName=trim(oRs.Fields("FirstName"))
```

```
    strOtherInits=trim(oRs.Fields("OtherInits"))
    strSurname=trim(oRs.Fields("Surname"))
    strWebAddress=trim(oRs.Fields("WebAddress"))
    strAffiliation=trim(oRs.Fields("Affiliation"))
    strDisplayName=trim(oRs.Fields("DisplayName"))

%>
<!-- #include virtual="/jipam/common/SubScribe.inc" -->

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module ModLogon.Asp

Module Name – PDF.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>PDF</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="PDF"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<div align="left">
  <table border="0" cellpadding="17" cellspacing="0" width="100%">
    <tr>
      <td width="100%" colspan="2">
        <h1 align="center"><font color="#FF0000">A</font>bout <font
color="#FF0000">P</font>DF
        <font color="#FF0000">F</font>iles</h1>
        All articles produced for JIPAM are in PDF format. Two PDF versions are
        supplied for each paper.<br>
        <br>
        <div align="left">
          <table border="1" cellpadding="9" cellspacing="0" width="100%">
            <tr>
              <td width="77%" valign="top" bgcolor="#FFFFCC">The first is
                designed to be viewed from a 15'' monitor. Where appropriate,
                the 'monitor' version will use colour and have links within the
                document and the WWW.</td>
              <td width="23%">
                <p align="center"><a href="images/001_99_www.pdf" target="_blank"><img
src="images/viewpanl.jpg" alt="Download paper suitable for viewing on screen"
border="0" width="50" height="71"></a></td>
            </tr>
            <tr>
              <td width="77%" valign="middle" bgcolor="#FFFFCC">The second is
                designed to be printed on an A4 page. To print, please use 'Fit
                to Page'.</td>
              <td width="23%">
                <p align="center"><strong><a href="images/001_99.pdf"
target="_blank"><img src="images/BS00883A.gif" width="79" height="74" alt="Download
paper suitable for printing" border="0"></a></strong></td>
            </tr>
            <tr>
              <td width="100%" valign="middle" bgcolor="#FFCCFF" colspan="2"><i>To
                view these files we recommend you save them to your file system
                and then view by using the Adobe Acrobat Reader. </i>
                <p><i>That is, click on the icon using the 2nd mouse button and
                select &quot;</i>Save Target As...&quot; (Microsoft Internet
                Explorer) <i>or</i> &quot;Save Link As...&quot; (Netscape
                Navigator).</p>
              </td>
            </tr>
          </table>
        </div>
        <h3><font color="#E60000"><big>W</big></font>hy <font
color="#E60000"><big>P</big></font>DF?</h3>
        <p>Mathematical papers are usually typeset in <a href="http://www-cs-
faculty.Stanford.EDU/~knuth/">Knuth's</a>
        <a href="http://www.tug.org">TeX</a> system, as are the articles in
        JIPAM<sup><a href="#note">*</a></sup>.  This produces DVI (device
        independent) files, a page description language that has only crude
        support for diagrams, graphics etc. The postscript language, which is
        popular for its power in combining text, graphics etc., results in files
        that are too large for transmission over the internet. PDF, a form of
        compressed postscript, has the capability of combining text and graphics
        as well as support for colour, links and web forms.</p>
        <p>To view PDF files, you'll need:</p>
      </td>
    </tr>
    <tr>
      <td width="60%">Adobe's PDF reader which is freely available from their
        web site: <a href="http://www.adobe.com/products/acrobat/readstep.html"
target="_blank">http://www.adobe.com/products/acrobat/readstep.html</a>
        <p><b>Note: </b><a href="http://www.ghostscript.com/">Ghostscript</a>
        (Ghostview) is not suitable for viewing the PDF files due to it's lack of
support for some of the PDF features we use.</p>
      </td>
      <td width="40%">
        <p align="center"><a href="http://www.adobe.com"><img src="images/logo.gif"
width="142" height="33" alt="logo.gif (897 bytes)"></a><br>
        <a href="http://www.adobe.com/products/acrobat/readstep.html"><img
src="images/mainsm.jpg" width="142" height="96" alt="mainsm.jpg (6342 bytes)"></a></p>
      </td>
    </tr>
    <tr>
      <td colspan="2" width="100%"> 
        <p><font size="2"><a name="note"></a>(*) JIPAM uses <a
href="http://thetis.fi.muni.cz/noauth/filist/fi_1.cgi.en?who=thanh">Hàn
        Thế Thành</a>'s <a href="http://www.tug.org/applications/pdftex">PDFTeX</a>,
        which translates TeX directly to PDF.  Adobe Acrobat is then used
        for the final finishing touches.</font></p>
      </td>
    </tr>
  </table>
</div>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Module Name – Search.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Search Page for JIPAM</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="SEARCH"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
  Dim strSearch
  Dim intOption
  strSearch=""
  intOption=1
%>
<H1 ALIGN="center"><FONT
COLOR="#FF0000">S</FONT>earch <FONT COLOR="#FF0000">T</FONT>he
<FONT COLOR="#FF0000">J</FONT>IPAM <FONT COLOR="#FF0000">D</FONT>atabase
</H1>
<p></p>
<!-- #include virtual="/jipam/common/SEARCH.inc" -->
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

MODULE NAME – SEARCH.INC

```
<center>
<form METHOD="POST" ACTION="DoSearch.asp">
<p>Enter the text you would like to search for <input TYPE="TEXT"
NAME="TxtSearch"
Value="<%=strSearch%>"></p>
<p>Where would you like the search performed</p>
<table WIDTH="50%">
<tr>
<% if intOption=1 then %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="1" CHECKED="CHECKED">Author</td>
<% else %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="1" >Author</td>
<% end if %>
</tr>
<tr>
<% if intOption=2 then %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="2" CHECKED="CHECKED">Title</td>
<% else %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="2">Title</td>
<% end if %>
</tr>
<tr>
<% if intOption=3 then %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="3"
CHECKED="CHECKED">Keyword</td>
<% else %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="3">Keyword</td>
<% end if %>
</tr>
<tr>
<% if intOption=4 then %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="4" CHECKED="CHECKED">Math
Code</td>
<% else %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="4">Math Code</td>
<% end if %>
</tr>
<tr>
<% if intOption=5 then %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="5"
CHECKED="CHECKED">Abstract</td>
<% else %>
<td> <input TYPE="RADIO" NAME="RdoSearch" VALUE="5">Abstract</td>
<% end if %>
</tr>
</table>
<p><input TYPE="SUBMIT" NAME="Submit1"></p>
</form>
<p></p>
<p></p>
</center>
```

Module Name – ShowArtImg.asp

```
<%@ Language=VBScript %>
<!-- #include virtual="/jipam/common/common.inc" -->
<!-- #include virtual="/jipam/common/error.inc" -->
<% ' this process returns a binary file
  Dim PageName
  PageName="NONAME"

  Dim strArticleId
  Dim iImageNumber
  strArticleId  = Request.QueryString("AI")
  iImageNumber = Request.QueryString("IM")

  On Error Resume next
  'Step 2: grab the picture from the database
  Dim oConn,OCmd, oRs
   'create and open a connection to the database
   Set oConn = Server.CreateObject("ADODB.Connection")
   if Err.number > 0 then Response.End

   oConn.Open strConnection
   if Err.number > 0 then Response.End

   'create a command and set the properties
   Set oCmd = Server.CreateObject("ADODB.Command")
   if Err.number > 0 then Response.End
   oCmd.ActiveConnection = oConn
   oCmd.CommandType = 4           'stored procedure
   oCmd.CommandText = "GetArticleImage"   'procedure name
   lngRecsAffected=0
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute(lngRecsAffected, Array(strArticleId, iImageNumber))
   if Err.number > 0 then Response.End
   if oRs.BOF then
     Response.end
   end if

   strImageType = oRs.Fields("ImageType")

   Set adoFldBlob=oRs("AnImage")
   lngFieldDataLength=adoFldBlob.ActualSize
   lngBlockCount=lngFieldDataLength / 4096
   lngRemainingData=lngFieldDataLength mod 4096

   Response.ContentType = "image/" + strImageType
   for lngCounter=1 to lngBlockCount
      Response.BinaryWrite (adoFldBlob.GetChunk(4096))
   next
   if lngRemainingData > 0 then
       Response.BinaryWrite (adoFldBlob.GetChunk(lngRemainingData))
   end if
   adoFldBlob.Close
   Set adoFldBlob=nothing

  'Clean up...
  oRS.Close
  Set oRS = Nothing
  Set oCmd = Nothing
  oCon.Close
  Set oCon = Nothing
%>
```

MODULE NAME – SHOWLOG.ASP

```
<%@ Language=VBScript %>
<%
QUOT = Chr(34)
strQuery = Request.Form("query")
strCriteria = Request.Form("criteria")
strWhere = ""
strSessW = ""

If Len(strCriteria) > 1 Then
   strSessW = " WHERE " & strCriteria
End If

Select Case strQuery
  Case "refer"
    strSQL = "SELECT Count_=SUM(ItemCount), Referring_Page=MAX(ItemText) FROM
RefererSummary" & strSessW & " GROUP BY ItemText ORDER BY SUM(ItemCount) DESC"
  Case "refcount"
    strSQL = "SELECT Week=MAX(TWeekNumber), Year=MAX(TYearNumber),
Referrals=SUM(ItemCount) FROM RefererSummary" & strSessW & " GROUP BY TYearNumber,
TWeekNumber ORDER BY MAX(TYearNumber) DESC, MAX(TWeekNumber) DESC"
  Case "reftarg"
    strSQL = "SELECT Hits=SUM(ItemCount), Target_Page=MAX(ItemText) FROM
SessionTargetSummary" & strSessW & " GROUP BY ItemText ORDER BY SUM(ItemCount) DESC"
  Case "agent"
    strSQL = "SELECT Hits=SUM(ItemCount), UserAgent_Type=MAX(CASE WHEN CHARINDEX(')',
ItemText) > 10 THEN SUBSTRING(ItemText, 1, CHARINDEX(')', ItemText)) ELSE ItemText
END) FROM UserAgentSummary" & strSessW & " GROUP BY CASE WHEN CHARINDEX(')', ItemText)
> 10 THEN SUBSTRING(ItemText, 1, CHARINDEX(')', ItemText)) ELSE ItemText END ORDER BY
SUM(ItemCount) DESC"
  Case "agentvia"
    strSQL = "SELECT Hits=SUM(ItemCount), UserAgent_Type=MAX(ItemText) FROM
UserAgentSummary" & strSessW & " GROUP BY ItemText ORDER BY SUM(ItemCount) DESC"
  Case "lang"
    strSQL = "SELECT Hits=SUM(ItemCount), Language=MAX(ItemText) FROM CountrySummary"
& strSessW & " GROUP BY ItemText ORDER BY SUM(ItemCount) DESC"
End Select
%>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<!-- #include virtual="/jipam/common/error.inc" -->
<HEAD>
<TITLE>Show Log</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="TRAFFICLOG"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
   if Not IsAdmin() or len(strQuery) < 1 then
      Response.Write("<B>Sorry, this function is not available </B>")
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
    Response.End
   end if
%>
<H1 ALIGN="center">
<% Select Case strQuery
   case "agent" %>
<FONT COLOR="#FF0000">B</FONT>rowser
<FONT COLOR="#FF0000">T</FONT>ypes
<% case "agentvia" %>
<FONT COLOR="#FF0000">B</FONT>rowser
<FONT COLOR="#FF0000">T</FONT>ypes (including via)
<% case "lang" %>
<FONT COLOR="#FF0000">U</FONT>ser
<FONT COLOR="#FF0000">A</FONT>gent
<FONT COLOR="#FF0000">L</FONT>anguages
<% case "refer" %>
<FONT COLOR="#FF0000">R</FONT>eferrer
<FONT COLOR="#FF0000">C</FONT>lick-throughs
<% case "refcount" %>
<FONT COLOR="#FF0000">S</FONT>ummary
<FONT COLOR="#FF0000">R</FONT>eferrer
<FONT COLOR="#FF0000">C</FONT>lick-throughs
<% case "reftarg" %>
<FONT COLOR="#FF0000">R</FONT>eferrer
<FONT COLOR="#FF0000">T</FONT>argets
<% end Select %>
</H1>

<%
If Request("showsql") = "on" Then
%>
<HR>Your query string was<P>
<%
  Response.Write strSQL & "<HR>"
End If

On Error Resume Next
Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open strConnLogs

Set oRs1 = oConn.Execute(strSQL)
If Err.Number > 0 Then
  Response.Write "<B>The database is empty or cannot be accessed at present</B>"
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
  Response.End
End If
If oRs1.EOF Then
  Response.Write "<B>Your query returned no records.</B>"
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
  Response.End
End If
%>

Return to <B><A HREF="TrafficReports.asp">Visitor Reports Menu</A> page.<P>

<%
oRs1.MoveFirst
  Response.Write "<TABLE><TR>" & vbCrlf
  For Each objItem In oRs1.Fields
    Response.Write "<TH NOWRAP ALIGN=" & QUOT & "LEFT" & QUOT & ">  " &
objItem.Name & "  </TH>" & vbCrlf
  Next
  Response.Write "</TR>" & vbCrlf
```

```
  Do While Not oRs1.EOF
    Response.Write "<TR>" & vbCrlf
    For Each objItem In oRs1.Fields
      Response.Write "<TD NOWRAP ALIGN=" & QUOT & "LEFT" & QUOT & ">  " &
objItem.value & "  </TD>" & vbCrlf
    Next
    Response.Write "</TR>" & vbCrlf
    oRs1.MoveNext
  Loop

Set oRs = Nothing
Set oRs1 = Nothing
Set oConn = Nothing
%>
</TABLE>
<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module ShowLog.asp

MODULE NAME – SUBSCRIBE.ASP

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Subscribe</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="SUBSCRIBE"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
Sub TerminateWithMessage(Mess)
   if len(Mess) > 0 then Response.Write(Mess)
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
  Response.End
end Sub
Function MakeSureNotNull(inputStr)
  if inputStr is null then
     MakeSureNotNull=""
  else
     MakeSureNotNull=inputStr
  end if
end Function
%>
<H1 ALIGN="left"><FONT
COLOR="#FF0000">S</FONT>ubscriptions
</H1>

<%
    'test to see if we are calling the form directly
    if len(Session("UserLogon")) > 0 then
       Mess = "<BR><BR> Sorry, You are already logged on as " + Session("UserLogon") +
"<P></P><HR>"
       TerminateWithMessage Mess
    end if
    'test to see if we are calling the form directly after being locked
    if Session("Locked") then
       Mess = "<BR><BR> Sorry, You are have been locked out of logging in for your
current session.<P></P>"
       TerminateWithMessage Mess
    end if
%>
<% 'get values out of the form as the form calls itset
  On Error Resume Next
  Dim strEMailAddress
  Dim strPassWord
  Dim strFirstName
  Dim strOtherInits
  Dim strSurname
  Dim strWebAddress
  Dim strAffiliation
  Dim strDisplayName 'NB this field displays only on modlogon.asp
  Dim intResult
  strEMailAddress=""
  strPassWord=""
  strFirstName=""
  strOtherInits=""
  strSurname=""
  strWebAddress=""
  strAffiliation=""
  strDisplayName=""
  intResult=0
  if Request.Form.Count > 1 then 'this is the a second or greater pass
    if Session("LoginCount") < 0 then
       Session("Locked")= true
       Mess = "<BR><BR> Sorry, You are have been locked out due to exceeding the
number of login attemps.<P></P>"
       TerminateWithMessage Mess
    end if
    Session("LoginCount") = Session("LoginCount") - 1
    strEMailAddress = Request.Form.Item("E1")
    strPassWord = Request.Form.Item("E2")
    strFirstName = MakeSureNotNull(Request.Form.Item("E4"))
    strOtherInits = MakeSureNotNull(Request.Form.Item("E5"))
    strSurname = MakeSureNotNull(Request.Form.Item("E6"))
    strWebAddress = MakeSureNotNull(Request.Form.Item("E7"))
    strAffiliation = MakeSureNotNull(Request.Form.Item("E8"))
    intUser=0 'default for all initial settings
    'create and open a connection to the database
    Err.Clear()
    Set oCon = Server.CreateObject("ADODB.Connection")
    if Err.number > 0 then  TerminateWithMessage "Sorry, Cannot connect to database(1)
"
    oCon.Open strConnection
    if Err.number > 0 then  TerminateWithMessage "Sorry, Cannot connect to database(2)
"
    Err.Clear()
   'create a command and set the properties
    Set oCmd = Server.CreateObject("ADODB.Command")
    if Err.number > 0 then  TerminateWithMessage "Sorry, Cannot connect to database(3)
"
    oCmd.ActiveConnection = oCon
    oCmd.CommandType = 4           'stored procedure
    oCmd.CommandText = "AddUserFull"   'procedure name
    lngRecsAffected=0
    arrParams=Array(strEmailAddress,
strPassWord,strFirstName,strOtherInits,strSurname,strWebAddress,strAffiliation,intUser
)
   'execute the command, supplying the parameters, and get the result
   Set oRs = oCmd.Execute(lngRecsAffected,arrParams )
   if Err.number > 0 then  TerminateWithMessage "Sorry, Cannot execute database
procedure"
   if not ORs.Bof then ' successful login
   ' int result contains the user_id value
     Session("UserLogon")=strEMailAddress
     Session("UseriD")=Ors.Fields("User_ID")
     Session("UsePass")=strPassWord 'Kept as parameter for procedures
GetUserDetails,and ModUsers
     Session("UserRole")=oRs.Fields("UserRole") ' this is default for all new users
     Mess="Thank you for registering. When logging on in future please use your E-Mail
address " + Session("UserLogon") +"<P></P>"
     TerminateWithMessage Mess
     Response.End
   else
     intResult=1
   end if
'we fall out here if the email address is in use
' intResult > 0 for this scenario
```

APPENDIX C – MODULE SUBSCRIBE.ASP

```
end if
%>
<%
   if intResult < 1 then 'a new subscription
%>
<P>While access to JIPAM is current free, you must subscribe to download full text
articles as PDF files. To subscribe to JIPAM for twelve months, complete the
subscription form
(areas marked * must be completed).</P><BR><BR>
<HR>
<%
  else
%>
<P>The E-Mail address you entered is already registered. If you have previously
registered please attempt to <A href="login.asp">login</A> instead. </P>
<%
  end if
%>

<!-- #include virtual="/jipam/common/SubScribe.inc" -->

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C - Module Subscribe.asp

MODULE NAME – SUBSCRIBE.INC

```
<SCRIPT LANGUAGE="JAVASCRIPT">
function validatePrompt(ctrl,mess) {
  alert(mess);
  ctrl.focus();
  return;
}
function ValidateEmail(form){
  Ctrl = form.E1;
  off = Ctrl.value.indexOf('@',0)
  if (Ctrl.value =="" || off < 2) {
  validatePrompt(Ctrl,"Enter a valid email address")
  return(false);
  }

  return (true);

}
function ValidatePass(form) {
  p1 = form.E2.value;
  p2 = form.E3.value;
  if (p1 != p2) {
    validatePrompt(form.E2,"Your password are not the same")
    return(false);
  }
  return (true);

}
function ValidateAllEntries(form) {
  if (!ValidateEmail(form)) return;
  if (!ValidatePass(form)) return;
  form.submit();
  return;
}
</SCRIPT>

<FORM METHOD="post" ACTION="<%=PageName%>.asp" NAME="Register">
<P></P>
<CENTER>
<TABLE WIDTH="90%">
<TR>
<TD WIDTH="171"> E-Mail Address *</TD>
<TD WIDTH="343"><INPUT NAME="E1" SIZE="50" MAXLENGTH="128"
VALUE="<%=strEMailAddress%>"></TD>
</TR>
<TR>
<TD WIDTH="171"> Password *</TD>
<TD WIDTH="343"><INPUT TYPE="password" NAME="E2" SIZE="15" MAXLENGTH="32"
VALUE="<%=strPassWord%>"></TD>
</TR>
<TR>
<TD WIDTH="171"> Password Again *</TD>
<TD WIDTH="343"><INPUT TYPE="password" NAME="E3" SIZE="15" MAXLENGTH="32"
VALUE="<%=strPassWord%>"></TD>
</TR>
<TR>
<TD WIDTH="171"> First Name</TD>
<TD WIDTH="343"><INPUT NAME="E4" SIZE="30" MAXLENGTH="60"
VALUE="<%=strFirstName%>"></TD>
</TR>
<TR>
<TD WIDTH="171"> Other Initials</TD>
<TD WIDTH="343"><INPUT NAME="E5" SIZE="15" MAXLENGTH="20"
VALUE="<%=strOtherInits%>"></TD>
</TR>
<TR>
<TD WIDTH="171"> Surname</TD>
<TD WIDTH="343"><INPUT NAME="E6" SIZE="50" MAXLENGTH="128"
VALUE="<%=strSurname%>"></TD>
</TR>
<TR>
<TD WIDTH="171"> Your Web address</TD>
<TD WIDTH="343"><INPUT NAME="E7" SIZE="50" MAXLENGTH="128"
VALUE="<%=strWebAddress%>"></TD>
</TR>
<TR>
<TD WIDTH="171"> Institution/Affiliation</TD>
<TD WIDTH="343"><INPUT NAME="E8" SIZE="35" MAXLENGTH="60"
VALUE="<%=strAffiliation%>"></TD>
</TR>
<%
   if Session("UserRole") > 0 then
%>
<TR>
<TD WIDTH="171"> Your Display Name</TD>
<TD WIDTH="343"><INPUT NAME="E9" SIZE="50" MAXLENGTH="128"
VALUE="<%=strDisplayName%>"></TD>
</TR>
<%
   end if
%>
<TR>
<TD WIDTH="171"> </TD>
<TD WIDTH="343"> </TD>
</TR>
<TR>
<TD WIDTH="171"> <INPUT TYPE="button" NAME="button1" VALUE="Submit Form"
ONCLICK="ValidateAllEntries(this.form)"></TD>
<TD WIDTH="343"> <INPUT TYPE="reset" NAME="Reset1"
VALUE="Reset Form"></TD>
</TR>
</TABLE>
</CENTER>
</FORM>
```

APPENDIX C – MODULE SUBSCRIBE.INC

Module Name – TrafficReports.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Traffic Report</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="TRAFFIC"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<%
   if Not IsAdmin() then
      Response.Write("<B>Sorry, this function is not available </B>")
%>
<!-- #include virtual="/jipam/common/Footer.inc" -->
<%
    Response.End
   end if
%>
<H1 ALIGN="center">
<FONT COLOR="#FF0000">V</FONT>isitor <FONT COLOR="#FF0000">R</FONT>eport
<FONT COLOR="#FF0000">M</FONT>enu
</H1>
<TABLE CELLPADDING="0" BORDER="0">
<FORM NAME="controls">
 <TR>
  <TD ALIGN="RIGHT" NOWRAP><INPUT TYPE="CHECKBOX" NAME="chkPeriod"
ONCLICK="clickPeriod()"></TD>
  <TD NOWRAP>For this period:</TD>
  </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="RIGHT" VALIGN="MIDDLE" NOWRAP>
   From week: <INPUT TYPE="TEXT" NAME="txtFromWeek" DISABLED VALUE="<% =
DatePart("ww", Now) - 1 %>" SIZE="2"> 
   <INPUT TYPE="TEXT" NAME="txtFromYear" DISABLED VALUE="<% = DatePart("yyyy", Now)
%>" SIZE="4"><BR>
   To week: <INPUT TYPE="TEXT" NAME="txtToWeek" DISABLED VALUE="<% = DatePart("ww",
Now) - 1 %>" SIZE="2"> 
   <INPUT TYPE="TEXT" NAME="txtToYear" DISABLED VALUE="<% = DatePart("yyyy", Now) %>"
SIZE="4">
  </TD>
 </TR>
 <TR>
  <TD COLSPAN="2"> </td>
 </TR>
 <TR>
  <TD COLSPAN="2"> </td>
 </TR>
 <TR>
  <TD COLSPAN="2">Report to run:</td>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="1" CHECKED > Breakdown of Browser Types
(chart in IE4 or higher)
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="2" > Summary of Browser Types
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="3" > Summary of Browser Types including
'via' details
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="4" > Breakdown of User Operating
Systems (chart in IE4 or higher)
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="5" > Breakdown of User Agent Languages
(chart in IE4 or higher)
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="6" > Summary of User Agent Languages
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="7" > Listing of Referrer Click-throughs
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="8" > Weekly Summary of Referrer Click-
throughs
  </TD>
 </TR>
 <TR>
  <TD></TD>
  <TD ALIGN="LEFT" NOWRAP>
   <INPUT TYPE="RADIO" NAME="optReport" VALUE="9" > Listing of Referrer Target URLs
  </TD>
 </TR>
</FORM>

 <TR>
 <TD> <FORM NAME="QryG1" ACTION="graphicuatype.asp" METHOD="POST">
      <INPUT TYPE="HIDDEN" NAME="criteria">
      <INPUT TYPE="HIDDEN" NAME="display">
      </FORM>
      <FORM NAME="QryG2" ACTION="graphicopsys.asp" METHOD="POST">
      <INPUT TYPE="HIDDEN" NAME="criteria">
```

Appendix C – Module TrafficReports.asp

```
      <INPUT TYPE="HIDDEN" NAME="display">
      </FORM>
      <FORM NAME="QryG3" ACTION="graphiclang.asp" METHOD="POST">
      <INPUT TYPE="HIDDEN" NAME="criteria">
      <INPUT TYPE="HIDDEN" NAME="display">
      </FORM>
   </TD>
  </TR>

<FORM NAME="QueryForm" ACTION="showlog.asp" METHOD="POST">
  <TR>
   <TD ALIGN="RIGHT" NOWRAP><INPUT TYPE="CHECKBOX" NAME="showsql"></TD>
   <TD ALIGN="LEFT" NOWRAP>Show SQL Query</TD>
  </TR>
  <TR>
   <TD COLSPAN="2"> </TD>
  </TR>
  <TR>
   <TD> </TD>
   <TD ALIGN="LEFT" NOWRAP COLSPAN="2">
     <INPUT TYPE="HIDDEN" NAME="criteria">
     <INPUT TYPE="HIDDEN" NAME="query">
     <INPUT TYPE="BUTTON" VALUE="Run Query" ONCLICK="runQuery()">  
   </TD>
  </TR>
</FORM>
</TABLE>
<SCRIPT LANGUAGE="JavaScript">
var objCForm = document.forms["controls"];
var objQForm = document.forms["QueryForm"];

function clickPeriod() {
  objCForm.txtFromWeek.disabled = !(objCForm.chkPeriod.checked);
  objCForm.txtFromYear.disabled = !(objCForm.chkPeriod.checked);
  objCForm.txtToWeek.disabled = !(objCForm.chkPeriod.checked);
  objCForm.txtToYear.disabled = !(objCForm.chkPeriod.checked)
}

function runQuery() {
  strCriteria = " ";
  strDisplay=" ";
  if (objCForm.chkPeriod.checked) {
     strCriteria = " (TYEARNUMBER + TWEEKNUMBER) >= (" + objCForm.txtFromYear.value  +
" + " + objCForm.txtFromWeek.value + ") AND "
     strCriteria += "(TYEARNUMBER + TWEEKNUMBER) <= (" + objCForm.txtToYear.value  + "
+ " + objCForm.txtToWeek.value + ")"
     strDisplay = "between week " + objCForm.txtFromWeek.value + " of " +
objCForm.txtFromYear.value;
     strDisplay += " and week " + objCForm.txtToWeek.value + " of " +
objCForm.txtToYear.value;
  } else {
     strCriteria = "(TYEARNUMBER + TWEEKNUMBER) = (" + objCForm.txtToYear.value  + " +
" + objCForm.txtToWeek.value + ")"
     strDisplay = " in week " + objCForm.txtToWeek.value + " of " +
objCForm.txtToYear.value;
  };
  actString="";
  if (objCForm.optReport[0].checked){
     document.forms["QryG1"].criteria.value=strCriteria
     document.forms["QryG1"].display.value =strDisplay
     document.forms["QryG1"].submit();
     return;
  }
  if (objCForm.optReport[1].checked) actString="agent";
  if (objCForm.optReport[2].checked) actString="agentvia";
  if (objCForm.optReport[3].checked){
     document.forms["QryG2"].criteria.value=strCriteria
     document.forms["QryG2"].display.value =strDisplay
     document.forms["QryG2"].submit();
     return;
  }
  if (objCForm.optReport[4].checked){
     document.forms["QryG3"].criteria.value=strCriteria
     document.forms["QryG3"].display.value =strDisplay
     document.forms["QryG3"].submit();
     return;
  }
  if (objCForm.optReport[5].checked) actString="lang";
  if (objCForm.optReport[6].checked) actString="refer";
  if (objCForm.optReport[7].checked) actString="refcount";
  if (objCForm.optReport[8].checked) actString="reftarg";

  objQForm.Action = actString;
  objQForm.criteria.value = strCriteria;
  objQForm.query.value = actString;

  objQForm.submit()
}
</SCRIPT>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix C – Module TrafficReports.asp

Module Name – WhatsNew.asp

```
<%@ Language=VBScript %>
<HTML>
<meta NAME="robots" CONTENT="noindex,nofollow">
<!-- #include virtual="/jipam/common/common.inc" -->
<HEAD>
<TITLE>Whats New</TITLE>
</HEAD>
<%
  Dim PageName
  PageName="WHATSNEW"
%>
<!-- #include virtual="/jipam/common/Mainheader.inc" -->
<!-- #include virtual="/jipam/common/menu.inc" -->
<I>Text for Whats New</I>

<!-- #include virtual="/jipam/common/Footer.inc" -->
```

Appendix D

JIPAM WEB SITE
Dynamic Database System

Administration Program
C++ Code

Module Name - DlgAddArtToVolume

DlgAddArtToVolume.h

```
//---------------------------------------------------------------------------
#ifndef DlgAddArtToVolumeH
#define DlgAddArtToVolumeH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <ComCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
//---------------------------------------------------------------------------
class TFrmAddArtToVolume : public TForm
{
__published:
        TQuery *QryGeneral;
        TPanel *Panel1;
        TListBox *LbxAssigned;
        TLabel *Label1;
        TPanel *Panel2;
        TListBox *LbxAvailable;
        TLabel *Label3;
        TPanel *Panel3;
        TButton *OKBtn;
        TButton *CancelBtn;
        TBitBtn *BtnDownAssigned;
        TBitBtn *BtnDownAvailable;
        TBitBtn *BtnUpAssigned;
        TBitBtn *BtnUpAvailable;
        void __fastcall OKBtnClick(TObject *Sender);
        void __fastcall LbxAssignedClick(TObject *Sender);
        void __fastcall BtnUpAssignedClick(TObject *Sender);
        void __fastcall BtnDownAssignedClick(TObject *Sender);
        void __fastcall BtnUpAvailableClick(TObject *Sender);
        void __fastcall BtnDownAvailableClick(TObject *Sender);
private:
    AnsiString DatabaseName;
    int Volume_ID;
    void __fastcall ClearObjects(TListBox*);
    void __fastcall LoadListBox(TListBox* tgtlbx);
    void __fastcall FixPositionIndex();
public:
      virtual __fastcall TFrmAddArtToVolume(TComponent* AOwner);
       virtual __fastcall ~TFrmAddArtToVolume();

        void __fastcall SetUp(const AnsiString dbname,
                int volume_id);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmAddArtToVolume *FrmAddArtToVolume;
//---------------------------------------------------------------------------
#endif
```

DlgAddArtToVolume.cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgAddArtToVolume.h"

// NB I have used the field institution in the TMObject to indicate
// whether the article has been moved between listboxes
// If it has been moved then the record behind it will need to be updated

//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmAddArtToVolume *FrmAddArtToVolume;
//---------------------------------------------------------------------------
__fastcall TFrmAddArtToVolume::TFrmAddArtToVolume(TComponent* AOwner)
      : TForm(AOwner)
{
}
//---------------------------------------------------------------------------
__fastcall TFrmAddArtToVolume::~TFrmAddArtToVolume(){
   ClearObjects(LbxAssigned);
   ClearObjects(LbxAvailable);
}
//---------------------------------------------------------------------------
void __fastcall TFrmAddArtToVolume::ClearObjects(TListBox* tlb){
  for (int i=0; i<tlb->Items->Count; i++){
     ArticleObj* tob = (ArticleObj*) tlb->Items->Objects[i];
     delete tob;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFrmAddArtToVolume::SetUp(const AnsiString dbname, int volume_id){
    DatabaseName=dbname;
    Volume_ID= volume_id;
    QryGeneral->DatabaseName= dbname;
    LoadListBox(LbxAssigned);
    LoadListBox(LbxAvailable);
    FixPositionIndex();
}
//---------------------------------------------------------------------------
void __fastcall TFrmAddArtToVolume::LoadListBox(TListBox* tgtlbx){

   AnsiString Query;
   if (tgtlbx == LbxAvailable)
     Query="Select article_Id,ArticleTitle from Articles where Published='N' and
Available='Y'";
   else {
     Query="Select article_Id,ArticleTitle,ArticleNO from TitlesInVolume where
Volume_iD=";
     Query+= IntToStr(Volume_ID);
     Query+= " order by ArticleNO";
   }
    QryGeneral->Close();
    QryGeneral->SQL->Clear();
    QryGeneral->SQL->Add(Query);
    try {
    QryGeneral->Open();
      while (! QryGeneral->Eof) {
         AnsiString Title = QryGeneral->FieldByName("ArticleTitle")->AsString;
```

```
        // the following only applies to LbxAvailable but running it against
        // LbxAssigned does no harm so no work around code is required
        int off = LbxAssigned->Items->IndexOf(Title);
        if (off < 0) {
            ArticleObj* tob = new ArticleObj();
            tob->SetArticleId (
                QryGeneral->FieldByName("article_Id")->AsString);
            tgtlbx->Items->AddObject(Title,tob);
        }
        QryGeneral->Next();
      }
      QryGeneral->Close();
      if (tgtlbx->Items->Count > 0) tgtlbx->ItemIndex=0;
    } catch (Exception& e) {
      ShowMessage(e.Message);
      OKBtn->Enabled=false;
    }
}



void __fastcall TFrmAddArtToVolume::OKBtnClick(TObject *Sender)
{
//
    AnsiString Query;
    TSession* ts = Sessions->FindSession(QryGeneral->SessionName);
    TDatabase* td = ts->FindDatabase(DatabaseName);
    td->StartTransaction();
    try {
       Query = "Delete from VolumeArticles where Volume_id=";
       Query += IntToStr(Volume_ID);
       QryGeneral->Close();
       QryGeneral->SQL->Clear();
       QryGeneral->SQL->Add(Query);
       QryGeneral->ExecSQL();
       QryGeneral->SQL->Clear();
       QryGeneral->SQL->Add("Insert into
VolumeArticles(Article_ID,Volume_id,ArticleNO) ");
       QryGeneral->SQL->Add("VALUES (:Article_ID ,:Volume_id ,:ArticleNO)");
       QryGeneral->Prepare();
       QryGeneral->ParamByName("Volume_id")->AsInteger = Volume_ID;
       for (int i=0; i<LbxAssigned->Items->Count;i++){
         ArticleObj* tob = (ArticleObj*) LbxAssigned->Items->Objects[i];
         QryGeneral->ParamByName("Article_ID")->AsString = tob->GetArticleId();
         QryGeneral->ParamByName("ArticleNO")->AsInteger = i+1;
         QryGeneral->ExecSQL();
       }
       QryGeneral->UnPrepare();
// Now make sure that the articles are published
// NB I could also check the moved flag (instituion_id) to see whether
// or not an update is required; but this bulk approach ensures that
// the data is correct (There is probably only 12 or so records affected
       QryGeneral->SQL->Clear();
       QryGeneral->SQL->Add("Update Articles Set Published = 'Y',Available ='Y' ");
       QryGeneral->SQL->Add(" where Article_id = :Article_ID ");
       QryGeneral->Prepare();
       for (int i=0; i<LbxAssigned->Items->Count;i++){
         ArticleObj* tob = (ArticleObj*) LbxAssigned->Items->Objects[i];
         QryGeneral->ParamByName("Article_ID")->AsString = tob->GetArticleId();
         QryGeneral->ExecSQL();
       }
       QryGeneral->UnPrepare();
// Now make sure that if we unpublished any articles they also are flagged as such
       QryGeneral->SQL->Clear();
       QryGeneral->SQL->Add("Update Articles Set Published = 'N',Available ='Y' ");
       QryGeneral->SQL->Add(" where Article_id = :Article_ID ");
       QryGeneral->Prepare();
       for (int i=0; i<LbxAvailable->Items->Count;i++){
         ArticleObj* tob = (ArticleObj*) LbxAvailable->Items->Objects[i];
         if (tob->GetMoved() ) { // the record was moved
         QryGeneral->ParamByName("Article_ID")->AsString = tob->GetArticleId();
         QryGeneral->ExecSQL();
         }
       }
       QryGeneral->UnPrepare();
       td->Commit();
    } catch (Exception& e) {
      td->Rollback();
      ShowMessage(e.Message);
    }

}
//---------------------------------------------------------------------------
 void __fastcall TFrmAddArtToVolume::FixPositionIndex(){
 // adjust the movement buttons
   BtnUpAssigned->Enabled =false;
   BtnDownAssigned->Enabled=false;
   BtnDownAvailable->Enabled=false;
   BtnUpAvailable->Enabled =false;

   if (LbxAssigned->Items->Count > 0 && LbxAssigned->ItemIndex >=0 ) {
     BtnDownAvailable->Enabled= true;
     if (LbxAssigned->ItemIndex > 0) // if not first item
        BtnUpAssigned->Enabled = true;
     if (LbxAssigned->ItemIndex < LbxAssigned->Items->Count-1 ) //not last item
           BtnDownAssigned->Enabled= true;
   }
   if (LbxAvailable->Items->Count > 0 && LbxAvailable->ItemIndex >=0 ) {
     BtnUpAvailable->Enabled = true;
   }
};

void __fastcall TFrmAddArtToVolume::LbxAssignedClick(TObject *Sender)
{
  FixPositionIndex();
}
//---------------------------------------------------------------------------

void __fastcall TFrmAddArtToVolume::BtnUpAssignedClick(TObject *Sender)
{
// reorder current LbxItem up
   int indx=LbxAssigned->ItemIndex;
      // push current item up by one
     LbxAssigned->Items->Move(indx,indx-1);
     LbxAssigned->ItemIndex = indx-1;

   FixPositionIndex();
   OKBtn->Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmAddArtToVolume::BtnDownAssignedClick(TObject *Sender)
{
// reorder down 1
   int indx=LbxAssigned->ItemIndex;
     // push current item over by one
     LbxAssigned->Items->Move(indx,indx+1);
```

Appendix D – Module DlgAddArtToVolume

```
     LbxAssigned->ItemIndex = indx+1;
   FixPositionIndex();
   OKBtn->Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmAddArtToVolume::BtnUpAvailableClick(TObject *Sender)
{
// move from available to assigned
// move items between list boxes
   int indx;
   AnsiString old;
   ArticleObj* tob;

   indx = LbxAvailable->ItemIndex;
   old = LbxAvailable->Items->Strings[indx];
   tob = (ArticleObj*) LbxAvailable->Items->Objects[indx];
   tob->SetMoved(); // flag moved
   LbxAvailable->Items->Delete(indx);
   if (LbxAvailable->Items->Count-1 >= indx)
       LbxAvailable->ItemIndex =indx;
   else
       LbxAvailable->ItemIndex =indx-1;
     // put string in other box
   LbxAssigned->Items->AddObject(old,tob);
   LbxAssigned->ItemIndex =
       LbxAssigned->Items->IndexOf (old);

   FixPositionIndex();
   OKBtn->Enabled = true;

}
//---------------------------------------------------------------------------

void __fastcall TFrmAddArtToVolume::BtnDownAvailableClick(TObject *Sender)
{
// move from assigned to available
// move items between list boxes
   int indx;
   AnsiString old;
   ArticleObj* tob;

     indx=LbxAssigned->ItemIndex ;
     old = LbxAssigned->Items->Strings[indx];
     tob =  (ArticleObj*) LbxAssigned->Items->Objects[indx];
     tob->SetMoved(); // flag moved
     LbxAssigned->Items->Delete(indx);
     if (LbxAssigned->Items->Count-1 >= indx)
         LbxAssigned->ItemIndex =indx;
     else
         LbxAssigned->ItemIndex =indx-1;
     // put string in other box
     LbxAvailable->Items->AddObject(old,tob);
     LbxAvailable->ItemIndex =
        LbxAvailable->Items->IndexOf (old);

   FixPositionIndex();
   OKBtn->Enabled = true;

}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgAddArtToVolume

Module Name - DlgAddAuthArt

DlgAddAuthArt.h

```
//---------------------------------------------------------------------------
#ifndef DlgAddAuthArtH
#define DlgAddAuthArtH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <ComCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
#include <Menus.hpp>
//---------------------------------------------------------------------------
class TFrmDlgAddAuthArt : public TForm
{
__published:
        TQuery *QryGeneral;
        TPopupMenu *PmnuAssigned;
        TMenuItem *MnuChoseOther;
        TMenuItem *MnuChangeInstitution;
        TPanel *Panel1;
        TLabel *lable1;
        TListBox *LbxAssigned;
        TLabel *Label1;
        TListBox *LbxAvailable;
        TPanel *Panel2;
        TButton *OKBtn;
        TButton *BtnSelectother;
        TButton *BtnChangeInstitution;
        TButton *CancelBtn;
        TBitBtn *BtnDownAssigned;
        TBitBtn *BtnUpAssigned;
        TBitBtn *BtnUpAvailable;
        TBitBtn *BtnDownAvailable;
        void __fastcall OKBtnClick(TObject *Sender);
        void __fastcall MnuChoseOtherClick(TObject *Sender);
        void __fastcall MnuChangeInstitutionClick(TObject *Sender);
        void __fastcall PmnuAssignedPopup(TObject *Sender);
        void __fastcall BtnUpAssignedClick(TObject *Sender);
        void __fastcall BtnDownAssignedClick(TObject *Sender);
        void __fastcall BtnUpAvailableClick(TObject *Sender);
        void __fastcall BtnDownAvailableClick(TObject *Sender);
        void __fastcall LbxAssignedClick(TObject *Sender);
private:
    AnsiString SessionName;
    AnsiString DatabaseName;
    AnsiString Article_ID;
    void __fastcall ClearObjects(TListBox*);
    void __fastcall LoadListBox(TListBox* tgtlbx);
    void __fastcall FixPositionIndex();
public:
       virtual __fastcall TFrmDlgAddAuthArt(TComponent* AOwner);
        virtual __fastcall ~TFrmDlgAddAuthArt();
        void __fastcall SetUp(const AnsiString sessname,const AnsiString dbname,
             AnsiString article_id);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmDlgAddAuthArt *FrmDlgAddAuthArt;
//---------------------------------------------------------------------------
#endif
```

DlgAddAuthArt.cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgAddAuthArt.h"
#include "DlgSelectNewAuthor.h"
#include "DlgChangeAuthorInstitution.h"

//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmDlgAddAuthArt *FrmDlgAddAuthArt;
//---------------------------------------------------------------------------
__fastcall TFrmDlgAddAuthArt::TFrmDlgAddAuthArt(TComponent* AOwner)
      : TForm(AOwner)
{
}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::SetUp(const AnsiString sessname,
          const AnsiString dbname,AnsiString article_id){
    SessionName=sessname;
    DatabaseName=dbname;
    Article_ID= article_id;
    QryGeneral->DatabaseName= dbname;
    LoadListBox(LbxAssigned);
    LoadListBox(LbxAvailable);
    FixPositionIndex();
}
//---------------------------------------------------------------------------
__fastcall TFrmDlgAddAuthArt::~TFrmDlgAddAuthArt(){
   ClearObjects(LbxAssigned);
   ClearObjects(LbxAvailable);
}
//---------------------------------------------------------------------------
void __fastcall TFrmDlgAddAuthArt::ClearObjects(TListBox* tlb){
  for (int i=0; i<tlb->Items->Count; i++){
     TMObject* tob = (TMObject*) tlb->Items->Objects[i];
     delete tob;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFrmDlgAddAuthArt::LoadListBox(TListBox* tgtlbx){
   ClearObjects(tgtlbx);
   AnsiString Query;
   if (tgtlbx == LbxAvailable)
     Query="Select user_id,Surname,FirstName,OtherInits,Institution_id from Users
where UserRole in (2,4)";
   else {
     Query="Select user_id,Surname,FirstName,OtherInits,Institution_id from Authors
where Article_id='";
     Query+= Article_ID + "'";
```

```
    }
    QryGeneral->Close();
    QryGeneral->SQL->Clear();
    QryGeneral->SQL->Add(Query);
    try {
    QryGeneral->Open();
      while (! QryGeneral->Eof) {
         AnsiString Auth = QryGeneral->FieldByName("Surname")->AsString;
         Auth+= "," +  QryGeneral->FieldByName("FirstName")->AsString;
         Auth+= " " +  QryGeneral->FieldByName("OtherInits")->AsString;
         // the following only applies to LbxAvailable but running it against
         // LbxAssigned does no harm so no work around code is required
         int off = LbxAssigned->Items->IndexOf(Auth);
         if (off < 0) {
             TMObject* tob = new TMObject(
                 QryGeneral->FieldByName("User_id")->AsInteger,
                 QryGeneral->FieldByName("Institution_id")->AsInteger,
                 // the user role is not used on this dialog box
                 0);
             tgtlbx->Items->AddObject(Auth,tob);
         }
         QryGeneral->Next();
      }
      QryGeneral->Close();
    } catch (Exception& e) {
      ShowMessage(e.Message);
      OKBtn->Enabled=false;
    }

}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::OKBtnClick(TObject *Sender)
{
 // when we click ok
    AnsiString Query;
    TSession* ts = Sessions->FindSession(SessionName);
    TDatabase* td = ts->FindDatabase(DatabaseName);
    td->StartTransaction();
    try {
       Query = "Delete from ArticleAuthors where article_id='";
       Query += Article_ID +"'";
       QryGeneral->Close();
       QryGeneral->SQL->Clear();
       QryGeneral->SQL->Add(Query);
       QryGeneral->ExecSQL();
       QryGeneral->SQL->Clear();
       QryGeneral->SQL->Add("Insert into
ArticleAuthors(Article_ID,User_ID,AuthorOrder,Institution_id) ");
       QryGeneral->SQL->Add("VALUES (:Article_ID ,:User_ID ,:AuthorOrder,
:Institution_id )");
       QryGeneral->Prepare();
       QryGeneral->ParamByName("Article_ID")->AsString = Article_ID;
       for (int i=0; i<LbxAssigned->Items->Count;i++){
         TMObject* tob = (TMObject*) LbxAssigned->Items->Objects[i];
         QryGeneral->ParamByName("User_ID")->AsInteger = tob->GetUser_id();
         QryGeneral->ParamByName("AuthorOrder")->AsInteger = i+1;
         QryGeneral->ParamByName("Institution_id")->AsInteger = tob-
>GetInstitution_id();
         QryGeneral->ExecSQL();
       }
       QryGeneral->UnPrepare();
       td->Commit();
    } catch (Exception& e) {
      td->Rollback();
      ShowMessage(e.Message);
    }
}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::MnuChoseOtherClick(TObject *Sender)
{

   TFrmSelectNewAuthor * Dlg = new TFrmSelectNewAuthor(this);
   Dlg->SetUp(DatabaseName);
   if (Dlg->ShowModal() == mrOk) {
// add the new entry from the listbox
     if (Dlg->Successful ) {
       TMObject* tob = new TMObject((TMObject*) Dlg->LbxNewAuthors->Items->Objects[
                 Dlg->LbxNewAuthors->ItemIndex ]);
       AnsiString auth =  Dlg->LbxNewAuthors->Items->Strings[
                 Dlg->LbxNewAuthors->ItemIndex ];
       LbxAssigned->Items->AddObject(auth,tob);
       OKBtn->Enabled =true;
     }
   }
   delete Dlg;
}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::MnuChangeInstitutionClick(
      TObject *Sender)
{
   if (LbxAssigned->ItemIndex < 0 ) return;
   TFrmChangeAuthorInstitution * Dlg = new TFrmChangeAuthorInstitution(this);
   AnsiString author = LbxAssigned->Items->Strings[
      LbxAssigned->ItemIndex];
   AnsiString caption = "Change Institution( ";
   caption += author + ")";
   Dlg->Caption = caption;
   TMObject* tob = (TMObject*) LbxAssigned->Items->Objects[
      LbxAssigned->ItemIndex];
   Dlg->SetUp(DatabaseName,tob);
   if (Dlg->ShowModal() == mrOk) OKBtn->Enabled = true;
   delete Dlg;

}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::PmnuAssignedPopup(TObject *Sender)
{
   if ( LbxAssigned->ItemIndex < 0)
     MnuChangeInstitution->Enabled =false;
   else
     MnuChangeInstitution->Enabled =true;

}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::BtnUpAssignedClick(TObject *Sender)
{
///   Reassign up
   int indx=LbxAssigned->ItemIndex;
   LbxAssigned->Items->Move(indx,indx-1);
   LbxAssigned->ItemIndex = indx-1;
   FixPositionIndex();
   OKBtn->Enabled = true;
```

Appendix D – Module DlgAddAuthArt

```
}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::BtnDownAssignedClick(TObject *Sender)
{
// reassign down 1
   int indx=LbxAssigned->ItemIndex;
   // push current item over by one
   LbxAssigned->Items->Move(indx,indx+1);
   LbxAssigned->ItemIndex = indx+1;
   FixPositionIndex();
   OKBtn->Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::BtnUpAvailableClick(TObject *Sender)
{
// move from available to assigned
   int indx;
   AnsiString old;
   TMObject* tob;

   // moving from available to assigned
   indx = LbxAvailable->ItemIndex;
   old = LbxAvailable->Items->Strings[indx];
   tob = (TMObject*) LbxAvailable->Items->Objects[indx];
   LbxAvailable->Items->Delete(indx);
   if (LbxAvailable->Items->Count-1 >= indx)
       LbxAvailable->ItemIndex =indx;
   else
       LbxAvailable->ItemIndex =indx-1;
     // put string in other box
   LbxAssigned->Items->AddObject(old,tob);
   LbxAssigned->ItemIndex =
       LbxAssigned->Items->IndexOf (old);
   FixPositionIndex();

   OKBtn->Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::BtnDownAvailableClick(TObject *Sender)
{
// move from assigned to available
   int indx;
   AnsiString old;
   TMObject* tob;

   indx=LbxAssigned->ItemIndex ;
   old = LbxAssigned->Items->Strings[indx];
   tob =  (TMObject*) LbxAssigned->Items->Objects[indx];
   LbxAssigned->Items->Delete(indx);
   if (LbxAssigned->Items->Count-1 >= indx)
       LbxAssigned->ItemIndex =indx;
   else
       LbxAssigned->ItemIndex =indx-1;
   // put string in other box
   LbxAvailable->Items->AddObject(old,tob);
   LbxAvailable->ItemIndex =
       LbxAvailable->Items->IndexOf (old);
   FixPositionIndex();
   OKBtn->Enabled = true;
}
//---------------------------------------------------------------------------
 void __fastcall TFrmDlgAddAuthArt::FixPositionIndex(){

 // adjust the movement buttons
   BtnUpAssigned->Enabled =false;
   BtnDownAssigned->Enabled=false;
   BtnDownAvailable->Enabled=false;
   BtnUpAvailable->Enabled =false;

   if (LbxAssigned->Items->Count > 0 && LbxAssigned->ItemIndex >=0 ) {
     BtnDownAvailable->Enabled= true;
     if (LbxAssigned->ItemIndex > 0) // if not first item
        BtnUpAssigned->Enabled = true;
     if (LbxAssigned->ItemIndex < LbxAssigned->Items->Count-1 ) //not last item
           BtnDownAssigned->Enabled= true;
   }
   if (LbxAvailable->Items->Count > 0 && LbxAvailable->ItemIndex >=0 ) {
     BtnUpAvailable->Enabled = true;
   }
};
//---------------------------------------------------------------------------

void __fastcall TFrmDlgAddAuthArt::LbxAssignedClick(TObject *Sender)
{
   FixPositionIndex();

}
//---------------------------------------------------------------------------
```

Module Name - DlgAddUser

DlgAddUser.h

```
//---------------------------------------------------------------------------
#ifndef DlgAddUserH
#define DlgAddUserH
//---------------------------------------------------------------------------
#include <vcl\ExtCtrls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\Classes.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Windows.hpp>
#include <vcl\System.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
#include <Mask.hpp>
//---------------------------------------------------------------------------
class TDlgFmAddUser : public TForm
{
__published:
        TButton *BtnAdd;
       TButton *CancelBtn;
       TBevel *Bevel1;
        TLabel *LBLEmail;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label3;
        TStoredProc *SprocAddUser;
        TLabel *LblError;
        TLabel *Label4;
        TLabel *Label5;
        TLabel *Label6;
        TLabel *Label7;
        TLabel *Label8;
        TMaskEdit *EdEmailAddress;
        TMaskEdit *EdPassword;
        TMaskEdit *EdFirstName;
        TMaskEdit *EdOtherInits;
        TMaskEdit *EdDisplayName;
        TMaskEdit *EdWebAddress;
        TMaskEdit *EdAffliation;
        TComboBox *CbxUserRole;
        TComboBox *CbxInstitution;
        TMaskEdit *EdSurname;
        TLabel *Label9;
        TLabel *Label10;
        TLabel *Label11;
        TQuery *QryGeneral;
        void __fastcall BtnAddClick(TObject *Sender);
        void __fastcall EdEmailAddressChange(TObject *Sender);
        void __fastcall CbxInstitutionChange(TObject *Sender);
private:
   int User_Id;
   bool SetUpInProgress;
   void InitEditFields();
public:
       virtual __fastcall TDlgFmAddUser(TComponent* AOwner);
        void SetUp(const AnsiString& DbName);
};
//---------------------------------------------------------------------------
extern PACKAGE TDlgFmAddUser *DlgFmAddUser;
//---------------------------------------------------------------------------
#endif
```

DlgAddUser.Cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgAddUser.h"
//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TDlgFmAddUser *DlgFmAddUser;
// Size Testor       = 12345678901234567890123456789012345678901234567890123456789012345678901234567890
//                   =          1         2         3         4         5         6         7         8
char* EDMPassword    ="CCCCccccccccccccccccccccccccccccc;1; "  ;
extern const int LengthEmailaddress=128;
extern const int LengthPassword = 32;
extern const int LengthFirstName=60;
extern const int LengthOtherInits=20;
extern const int LengthSurname=128;
extern const int LengthDisplayName=128;
extern const int LengthWebAddress=128;
extern const int LengthAffliation=60;

//---------------------------------------------------------------------------
__fastcall TDlgFmAddUser::TDlgFmAddUser(TComponent* AOwner)
      : TForm(AOwner)
{
  SetUpInProgress=true;
}
//---------------------------------------------------------------------------
void
TDlgFmAddUser::SetUp(const AnsiString& DbName){

   EdPassword->EditMask =   EDMPassword;
   EdEmailAddress->MaxLength=LengthEmailaddress;
   EdPassword->MaxLength=LengthPassword;
   EdFirstName->MaxLength=LengthFirstName;
   EdOtherInits->MaxLength=LengthOtherInits;
   EdSurname->MaxLength=LengthSurname;
   EdWebAddress->MaxLength=LengthWebAddress;
   EdAffliation->MaxLength=LengthAffliation;
   EdDisplayName->MaxLength=LengthDisplayName;
   SprocAddUser->DatabaseName=DbName;
   QryGeneral->DatabaseName=DbName;
   // load the institution list
   QryGeneral->SQL->Clear();
   QryGeneral->SQL->Add("Select SmallDesc from Institutions");
   QryGeneral->Open();
   while (!QryGeneral->Eof) {
     CbxInstitution->Items->Add(
       QryGeneral->FieldByName("SmallDesc")->AsString);
     QryGeneral->Next();
   }
   QryGeneral->Close();
   SetUpInProgress=false;
   InitEditFields();
```

```
}
//---------------------------------------------------------------------------


void __fastcall TDlgFmAddUser::BtnAddClick(TObject *Sender)
{
   LblError->Caption="";

   SprocAddUser->Params->Items[1]->AsString = EdEmailAddress->Text;
   SprocAddUser->Params->Items[2]->AsString = EdPassword->Text;
   SprocAddUser->Params->Items[3]->AsString = EdFirstName->Text;
   SprocAddUser->Params->Items[4]->AsString = EdOtherInits->Text;
   SprocAddUser->Params->Items[5]->AsString = EdSurname->Text;
   SprocAddUser->Params->Items[6]->AsString = EdWebAddress->Text;
   SprocAddUser->Params->Items[7]->AsString = EdAffliation->Text;
   SprocAddUser->Params->Items[8]->AsInteger = CbxUserRole->ItemIndex;
   //
   QryGeneral->SQL->Clear();
   AnsiString Query = "Select Institution_ID from Institutions where SmallDesc='";
   Query += CbxInstitution->Items->Strings[CbxInstitution->ItemIndex] + "'";
   QryGeneral->SQL->Add(Query);
   QryGeneral->Open();
   int instiD=QryGeneral->FieldByName("Institution_ID")->AsInteger;
   QryGeneral->Close();
   SprocAddUser->Params->Items[9]->AsInteger = instiD;
   SprocAddUser->Prepare();
   SprocAddUser->ExecProc();

   User_Id = SprocAddUser->Params->Items[0]->AsInteger;
   if (User_Id > 99) {
     LblError->Caption= "Added " + EdEmailAddress->Text + " as User " + User_Id;

   } else {
     LblError->Caption="Error - Email Address already exists"  ;
   }
}

//---------------------------------------------------------------------------

void __fastcall TDlgFmAddUser::EdEmailAddressChange(TObject *Sender)
{
  if (EdEmailAddress->Text.Length() >0) BtnAdd->Enabled=true;
  else                                  BtnAdd->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TDlgFmAddUser::CbxInstitutionChange(TObject *Sender)
{
   if (SetUpInProgress) return;
   EdAffliation->Text = CbxInstitution->Items->Strings[
        CbxInstitution->ItemIndex ];
}
//---------------------------------------------------------------------------
void
TDlgFmAddUser::InitEditFields(){
   EdEmailAddress->Text="";
   EdPassword->Text="";
   EdFirstName->Text="";
   EdOtherInits->Text="";
   EdSurname->Text="";
   EdWebAddress->Text="";
   EdAffliation->Text="NONE";
   EdDisplayName->Text="";
   CbxInstitution->Text="";
   LblError->Caption="";
   CbxUserRole->ItemIndex=0;
   CbxInstitution->ItemIndex=CbxInstitution->Items->IndexOf ("NONE");

}
//---------------------------------------------------------------------------
```

Module Name – DlgArticleDownloads

DlgArticleDownloads.h

```
//---------------------------------------------------------------------------
#ifndef DlgArticleDownloadsH
#define DlgArticleDownloadsH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
#include <Dialogs.hpp>
#include <ComCtrls.hpp>
#define ODBCVER 0x0300
#define ODBCVER 0x0300
#include <sqlext.h>
//---------------------------------------------------------------------------
class TDlgFrmDownloads : public TForm
{
__published:
        TButton *BtnAdd;
       TButton *CancelBtn;
       TBevel *Bevel1;
        TButton *BtnDelete;
        TListBox *LbxUsed;
        TOpenDialog *OpenDialog;
        TQuery *QryArticlesInFull;
        TListBox *LbxAvailable;
        TLabel *Label1;
        TLabel *Label2;
        TProgressBar *Progressbar;
        void __fastcall BtnAddClick(TObject *Sender);
        void __fastcall LbxUsedClick(TObject *Sender);
        void __fastcall LbxAvailableClick(TObject *Sender);
        void __fastcall BtnDeleteClick(TObject *Sender);
private:
    SQLHSTMT hstmt;
    AnsiString DatabaseName;
    AnsiString ArticleNo;
    void SetUpListBoxes();
    void __fastcall CheckStatementRetResult(SQLRETURN retcode );
    bool __fastcall WriteData(int filehandle,int PrintType);
    void __fastcall ClearObjects(TListBox* tlb);

public:
       virtual __fastcall TDlgFrmDownloads(TComponent* AOwner);
         virtual __fastcall ~TDlgFrmDownloads();
       void SetUp(AnsiString articelno,const AnsiString dbname);

};
//---------------------------------------------------------------------------
extern PACKAGE TDlgFrmDownloads *DlgFrmDownloads;
//---------------------------------------------------------------------------
#endif
```

DlgArticleDownloads.Cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "DlgArticleDownloads.h"
#include "JipamCommon.h"
#include <fcntl.h>
#include <io.h>

//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TDlgFrmDownloads *DlgFrmDownloads;
//---------------------------------------------------------------------------
__fastcall TDlgFrmDownloads::TDlgFrmDownloads(TComponent* AOwner)
      : TForm(AOwner)
{
  ArticleNo=0;
}
//---------------------------------------------------------------------------
__fastcall TDlgFrmDownloads::~TDlgFrmDownloads(){
   ClearObjects(LbxUsed);
   ClearObjects(LbxAvailable);

}

//---------------------------------------------------------------------------
void __fastcall TDlgFrmDownloads::ClearObjects(TListBox* tlb){
  for (int i=0; i<tlb->Items->Count; i++){
     PrintTypeObj* tob = (PrintTypeObj*) tlb->Items->Objects[i];
     delete tob;
  }
}
//---------------------------------------------------------------------------
void
TDlgFrmDownloads::SetUp(AnsiString articleno,const AnsiString dbname) {
  ArticleNo=articleno;
  DatabaseName=dbname;
  QryArticlesInFull->DatabaseName=dbname;
  SetUpListBoxes();
}
//---------------------------------------------------------------------------
void
TDlgFrmDownloads::SetUpListBoxes() {
  BtnDelete->Enabled = false;
  BtnAdd->Enabled=false;

  AnsiString Query =
    "Select PrintType,PrintShortDesc from PrintFileTypes "
     " where printType in ( "
  " Select PrintType from ArticlesInFull where Article_ID ='";
  Query += ArticleNo + "')" ;
  QryArticlesInFull->SQL->Clear();
  QryArticlesInFull->SQL->Add(Query);

  ClearObjects(LbxUsed);
  ClearObjects(LbxAvailable);
  LbxUsed->Clear();
  LbxAvailable->Clear();
  try {
```

```
    QryArticlesInFull->Open();
    while(!QryArticlesInFull->Eof) {
      PrintTypeObj* tob = new PrintTypeObj(
               QryArticlesInFull->FieldByName("PrintType")->AsInteger);
      LbxUsed->Items->AddObject(
         QryArticlesInFull->FieldByName("PrintShortDesc")->AsString,tob);

      QryArticlesInFull->Next();
    }
    QryArticlesInFull->Close();
  } catch (Exception& er1) {
    ShowMessage(er1.Message);
    return;
  }
  Query =
    "Select PrintType,PrintShortDesc from PrintFileTypes "
     " where printType not in ( "
  " Select PrintType from ArticlesInFull where Article_ID = '";
  Query += ArticleNo + "')" ;
  QryArticlesInFull->SQL->Clear();
  QryArticlesInFull->SQL->Add(Query);
  try {
     QryArticlesInFull->Open();
     while(!QryArticlesInFull->Eof) {
       PrintTypeObj* tob = new PrintTypeObj(
                QryArticlesInFull->FieldByName("PrintType")->AsInteger);
       LbxAvailable->Items->AddObject(
          QryArticlesInFull->FieldByName("PrintShortDesc")->AsString,tob);
       QryArticlesInFull->Next();
     }
     QryArticlesInFull->Close();
  } catch (Exception& er2) {
    ShowMessage(er2.Message);
    return;
  }

}
//---------------------------------------------------------------------------
void __fastcall TDlgFrmDownloads::BtnAddClick(TObject *Sender)
{
// add
   int PrintType;
   AnsiString filename;
   // get PrintType of item selected
   PrintTypeObj* tob = (PrintTypeObj*)  LbxAvailable->Items->Objects
                                          [LbxAvailable->ItemIndex];
   PrintType = tob->GetPrintType();

   if(OpenDialog->Execute()){
    filename = OpenDialog->FileName;
   }
   if (filename.Length () < 1) return; // no file selected
   try {
// lets find out the ODBC connection handle
     TSession* asess = Sessions->FindSession("SessionJipamA");
     TDatabase* adb = asess->FindDatabase(DatabaseName);

     SQLHANDLE hNativeDb;
     unsigned short Size;
     if (DbiGetProp(adb->Handle, dbNATIVEHNDL, &hNativeDb, sizeof(hNativeDb), Size)
          != DBIERR_NONE)
        throw Exception("Cannot access ODBC Handle");
// now obtain the statement handle
     if (SQLAllocHandle(SQL_HANDLE_STMT, hNativeDb, &hstmt) != SQL_SUCCESS)
        throw Exception("Cannot allocate statement Handle");
     int FileHandle = open(filename.c_str(),O_RDONLY | O_BINARY);
     if (FileHandle < 0) throw Exception(filename + " is unabled to be opened");
     TCursor temp = Screen->Cursor;
     Screen->Cursor = crHourGlass;
     try {
        WriteData(FileHandle,PrintType);
     } __finally {
        SQLFreeStmt(hstmt,SQL_CLOSE);
        close(FileHandle);
        Screen->Cursor = temp;
        Progressbar->Visible =false;
     }
   } catch (Exception & erl) {
      ShowMessage(erl.Message);
   }
   SetUpListBoxes();
}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmDownloads::LbxUsedClick(TObject *Sender)
{
  BtnDelete->Enabled = true;

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmDownloads::LbxAvailableClick(TObject *Sender)
{
  BtnAdd->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmDownloads::BtnDeleteClick(TObject *Sender)
{
   int PrintType;
   // get PrintType of item selected
   PrintTypeObj* tob = (PrintTypeObj*)  LbxUsed->Items->Objects
                                          [LbxUsed->ItemIndex];
   PrintType = tob->GetPrintType();

   AnsiString Query = "Delete from ArticlesInFull where Article_ID= '";
   Query += ArticleNo + "' and PrintType = " + IntToStr(PrintType);
   QryArticlesInFull->SQL->Clear();
   QryArticlesInFull->SQL->Add(Query);
   try {
     QryArticlesInFull->ExecSQL();
   } catch (Exception& erl) {
     ShowMessage(erl.Message);
     BtnDelete->Enabled = false;
     return;
   }
   SetUpListBoxes();
}
//---------------------------------------------------------------------------
void __fastcall
TDlgFrmDownloads::CheckStatementRetResult(SQLRETURN retcode ){
   SQLCHAR SqlState[6];
   SQLCHAR Msg[SQL_MAX_MESSAGE_LENGTH];
   SQLINTEGER NativeError;
   SQLSMALLINT MsgLen;

   if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO) ){
```

APPENDIX D – MODULE DLGARTICLEDOWNLOADS

```
      if (retcode == SQL_INVALID_HANDLE)
         throw Exception("OBDC Invalid handle");
      // error
      SQLGetDiagRec(SQL_HANDLE_STMT,hstmt,1,SqlState,&NativeError,Msg,
          sizeof(Msg),&MsgLen);
      throw Exception((char*)Msg);
   }
}
//---------------------------------------------------------------------------
bool
__fastcall TDlgFrmDownloads::WriteData(int filehandle,int PrintType) {

#define    MAX_DATA_LEN 4096
  char* Data[MAX_DATA_LEN];
  long FileLength = filelength(filehandle);
  Progressbar->Max = FileLength;
  Progressbar->Visible = true;
  Update();
  SQLINTEGER   cbPDFFILE=0;
  SQLPOINTER   pToken;
  SQLRETURN    retcode;

  char* Query = "INSERT INTO ARTICLESINFULL (ARTICLE_ID, PRINTTYPE, TheArticle) VALUES
(?, ?, ?)";


  retcode = SQLPrepare(hstmt,Query,SQL_NTS);
  if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO) )
      CheckStatementRetResult(retcode); // return failure

    /* Bind the parameters. For parameter 3, pass */
    /* the parameter number in ParameterValuePtr instead of a buffer */
    /* address.   */
    cbPDFFILE = SQL_LEN_DATA_AT_EXEC(FileLength);

    CheckStatementRetResult(SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR ,
                         SQL_CHAR, 6, 0, ArticleNo.c_str(), ArticleNo.Length(), 0));
    CheckStatementRetResult(SQLBindParameter(hstmt, 2, SQL_PARAM_INPUT, SQL_C_SHORT ,
                         SQL_SMALLINT, 0, 0, &PrintType, 0, 0));
    CheckStatementRetResult(SQLBindParameter(hstmt, 3, SQL_PARAM_INPUT,
                    SQL_C_BINARY, SQL_LONGVARBINARY,
                    0, 0, (SQLPOINTER) 3, 0, &cbPDFFILE));
    /* Set values so data for parameter 3 will be */
    /* passed at execution. Note that the length parameter in */
    /* the macro SQL_LEN_DATA_AT_EXEC is 0. This assumes that */
    /* the driver returns "N" for the SQL_NEED_LONG_DATA_LEN */
    /* information type in SQLGetInfo.
                          */

    retcode = SQLExecute(hstmt);
    if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO
         || retcode == SQL_NEED_DATA) )
      CheckStatementRetResult(retcode); // return failure

    /* For data-at-execution parameters, call SQLParamData to */
    /* get the parameter number set by SQLBindParameter. */
    /* Call InitUserData. Call GetUserData and SQLPutData */
    /* repeatedly to get and put all data for the parameter. */
    /* Call SQLParamData to finish processing this parameter */

    while (retcode == SQL_NEED_DATA) {
        retcode = SQLParamData(hstmt, &pToken);
        if (retcode == SQL_NEED_DATA) {
            SQLINTEGER cbData,DataLeft;
            DataLeft = FileLength;

            while ( true) {
               cbData = DataLeft > MAX_DATA_LEN ? MAX_DATA_LEN : DataLeft;
               if (read(filehandle,Data,cbData) < 0)
                  throw Exception("Unable to read from file");
               SQLPutData(hstmt, Data, cbData);
               DataLeft -= cbData;
               Progressbar->Position += cbData;
               Update();
               if (DataLeft <= 0) break;
            }

        }
    }
    CheckStatementRetResult(retcode); // test for failure
    return true; // flag that we were successful
}
```

Appendix D – Module DlgArticleDownloads

Module Name - DlgChangeAuthorInstitution

DlgChangeAuthorInstitution.h

```
//---------------------------------------------------------------------------
#ifndef DlgChangeAuthorInstitutionH
#define DlgChangeAuthorInstitutionH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
//---------------------------------------------------------------------------
class TFrmChangeAuthorInstitution : public TForm
{
__published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
          TCheckBox *CbxUpdate;
          TLabel *Label1;
          TListBox *LbxInstitution;
          TQuery *QryGeneral;
          TButton *BtnAddInstitution;
          void __fastcall LbxInstitutionClick(TObject *Sender);
          void __fastcall OKBtnClick(TObject *Sender);
          void __fastcall BtnAddInstitutionClick(TObject *Sender);
private:
       AnsiString DatabaseName;
       TMObject* TMob;
       void __fastcall ClearObjects(TListBox* tlb);
       void __fastcall SetUpListBox();
public:
        virtual __fastcall TFrmChangeAuthorInstitution(TComponent* AOwner);
          virtual __fastcall ~TFrmChangeAuthorInstitution();
          void __fastcall SetUp(const AnsiString dbname,TMObject* data);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmChangeAuthorInstitution *FrmChangeAuthorInstitution;
//---------------------------------------------------------------------------
#endif
```

DlgChangeAuthorInstitution.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgChangeAuthorInstitution.h"
#include "DlgModInstitution.h"

//---------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmChangeAuthorInstitution *FrmChangeAuthorInstitution;
//---------------------------------------------------------------------
__fastcall TFrmChangeAuthorInstitution::TFrmChangeAuthorInstitution(TComponent*
AOwner)
        : TForm(AOwner)
{
}
//---------------------------------------------------------------------
__fastcall TFrmChangeAuthorInstitution::~TFrmChangeAuthorInstitution(){
  ClearObjects(LbxInstitution);
}
//---------------------------------------------------------------------
void __fastcall TFrmChangeAuthorInstitution::SetUp(const AnsiString dbname,
                    TMObject* data){
     TMob=data;
     DatabaseName=dbname;
     QryGeneral->DatabaseName=dbname;
     QryGeneral->SQL->Add(
      "Select SmallDesc,Institution_ID from Institutions");
     SetUpListBox();

}
//---------------------------------------------------------------------
void __fastcall TFrmChangeAuthorInstitution::ClearObjects(TListBox* tlb){
  for (int i=0; i<tlb->Items->Count; i++){
     TMObject* tob = (TMObject*) tlb->Items->Objects[i];
     delete tob;
  }
}
//---------------------------------------------------------------------------

void __fastcall TFrmChangeAuthorInstitution::LbxInstitutionClick(
      TObject *Sender)
{
     OKBtn~>Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmChangeAuthorInstitution::OKBtnClick(TObject *Sender)
{
   AnsiString Query;
   TMObject* tob = (TMObject*) LbxInstitution->Items->Objects[
                          LbxInstitution->ItemIndex];
   if (tob->GetInstitution_id() == TMob->GetInstitution_id()) {
     // same as input so do nothing
     return;
   }
   if (CbxUpdate->Checked) { // need to update users for intituion id
      QryGeneral->Close();
      QryGeneral->SQL->Clear();
      Query = "Update Users set Institution_ID = ";
      Query += IntToStr(tob->GetInstitution_id());
      Query += " where user_id = ";
      Query += IntToStr(TMob->GetUser_id());
      try {
        QryGeneral->SQL->Add(Query);
        QryGeneral->ExecSQL ();
      } catch (Exception& e) {
        ShowMessage(e.Message);
      }
```

```
    }
    TMob->SetInstitution(tob->GetInstitution_id());
}
//---------------------------------------------------------------------------
void __fastcall TFrmChangeAuthorInstitution::SetUpListBox(){

    TMObject* tob;
    int institution_id;
    ClearObjects(LbxInstitution);
    LbxInstitution->Items->Clear();
    try {
      QryGeneral->Open();
      while (!QryGeneral->Eof) {
         institution_id=QryGeneral->FieldByName("Institution_ID")->AsInteger;
         tob = new TMObject(0,
            institution_id,0);
         LbxInstitution->Items->AddObject(
            QryGeneral->FieldByName("SmallDesc")->AsString,tob);
         QryGeneral->Next();
      }
      // set the item index for the current value
      for (int i=0;i< LbxInstitution->Items->Count ;i++){
         tob = (TMObject*) LbxInstitution->Items->Objects[i];
         if (tob->GetInstitution_id() == TMob->GetInstitution_id()) {
            LbxInstitution->ItemIndex = i;
            break;   // out of for loop
         }
      }
    } catch (Exception& e) {
      ShowMessage(e.Message);
      LbxInstitution->Enabled=false;
      OKBtn->Enabled =false;
    }
    QryGeneral->Close();
}
//---------------------------------------------------------------------------

void __fastcall TFrmChangeAuthorInstitution::BtnAddInstitutionClick(
      TObject *Sender)
{
   TFrmModInstitution* Dlg = new TFrmModInstitution(this);
   try {
      Dlg->SetUp(DatabaseName,false,0);
      if (Dlg->ShowModal() == mrOk) SetUpListBox();
      delete Dlg;
   } catch (Exception& e) {
      ShowMessage(e.Message);
      delete Dlg;
   }

}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgChangeAuthorInsitution

Module Name - DlgInstitutions

DlgInstitutions.h

```
//---------------------------------------------------------------------------
#ifndef DlgInstitutionsH
#define DlgInstitutionsH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBCtrls.hpp>
#include <DBGrids.hpp>
#include <DBTables.hpp>
#include <Grids.hpp>
//---------------------------------------------------------------------------
class TFrmInstitutions : public TForm
{
__published:
        TTable *TblInstitutions;
        TDataSource *DsInstituions;
        TQuery *QryGeneral;
        TPanel *Panel1;
        TButton *BtnAdd;
        TButton *BtnChange;
        TButton *BtnDelete;
        TButton *CancelBtn;
        TDBGrid *DBGridInstitutions;
        TDBNavigator *DBNavigator1;
        void __fastcall BtnAddClick(TObject *Sender);
        void __fastcall BtnChangeClick(TObject *Sender);
        void __fastcall BtnDeleteClick(TObject *Sender);
private:
     AnsiString DatabaseName;
public:
       virtual __fastcall TFrmInstitutions(TComponent* AOwner);
        void __fastcall SetUp(const AnsiString dbname);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmInstitutions *FrmInstitutions;
//---------------------------------------------------------------------------
#endif
```

DlgInstitutions.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "DlgInstitutions.h"
#include "DlgModInstitution.h"
//---------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmInstitutions *FrmInstitutions;
//---------------------------------------------------------------------
__fastcall TFrmInstitutions::TFrmInstitutions(TComponent* AOwner)
      : TForm(AOwner)
{
}
//---------------------------------------------------------------------
void __fastcall TFrmInstitutions::SetUp(const AnsiString dbname){
   DatabaseName=dbname;
   TblInstitutions->DatabaseName = dbname;
   QryGeneral->DatabaseName = dbname;
   TblInstitutions->Open();
}
//---------------------------------------------------------------------
void __fastcall TFrmInstitutions::BtnAddClick(TObject *Sender)
{
// add action
   TFrmModInstitution* Dlg=0;
   try {
     Dlg = new TFrmModInstitution(this);
     Dlg->SetUp(DatabaseName,false,0);
   } catch (Exception& e) {
     ShowMessage(e.Message);
     if (Dlg) delete Dlg;
     return;
   }
   if (Dlg->ShowModal() == mrOk) {
     TblInstitutions->Refresh();
     TblInstitutions->Last();
   };
   delete Dlg;
   DBGridInstitutions->Refresh();

}
//---------------------------------------------------------------------

void __fastcall TFrmInstitutions::BtnChangeClick(TObject *Sender)
{
// change button
   if (TblInstitutions->FieldByName("Institution_ID")->AsInteger == 1) {
     ShowMessage("Default institution cannot be changed");
     return;
   }
   TFrmModInstitution* Dlg=0;
   try {
     Dlg = new TFrmModInstitution(this);
     Dlg->EdSmallDesc->Text = TblInstitutions->FieldByName("SmallDesc")->AsString;
     Dlg->EdAddress1->Text = TblInstitutions->FieldByName("AddressLine1")->AsString;
     Dlg->EdAddress2->Text = TblInstitutions->FieldByName("AddressLine2")->AsString;
     Dlg->EdAddress3->Text = TblInstitutions->FieldByName("AddressLine3")->AsString;
     Dlg->EdAddress4->Text = TblInstitutions->FieldByName("AddressLine4")->AsString;
     Dlg->EdAddress5->Text = TblInstitutions->FieldByName("AddressLine5")->AsString;
     Dlg->EdTelephone->Text = TblInstitutions->FieldByName("Telephone")->AsString;
     Dlg->EdFascimile->Text = TblInstitutions->FieldByName("Facsimile")->AsString;
     Dlg->SetUp(DatabaseName,true,TblInstitutions->FieldByName("Institution_ID")->
>AsInteger);
   } catch (Exception& e) {
     ShowMessage(e.Message);
     if (Dlg) delete Dlg;
     return;
```

```
    }
    Dlg->ShowModal();
    delete Dlg;
      TblInstitutions->Refresh();
    DBGridInstitutions->Refresh();

}
//---------------------------------------------------------------------------

void __fastcall TFrmInstitutions::BtnDeleteClick(TObject *Sender)
{
// delete
    int instituionId = TblInstitutions->FieldByName("Institution_ID")->AsInteger;
    if (instituionId == 1) {
      ShowMessage("Default institution cannot be deleted");
      return;
    }
    try {
      QryGeneral->Close();
      QryGeneral->SQL->Clear();
      AnsiString Query="Delete from Institutions where Institution_ID=";
      Query += IntToStr(instituionId);
      QryGeneral->SQL->Add(Query);
      QryGeneral->ExecSQL();
      TblInstitutions->Refresh();
      DBGridInstitutions->Refresh();

    } catch (Exception& e) {
      ShowMessage(e.Message);
    }
}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgInstitutions

Module Name - DlgKeyWords

DlgKeyWords.h

```
//---------------------------------------------------------------------------
#ifndef DlgKeyWordsH
#define DlgKeyWordsH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
//---------------------------------------------------------------------------
class TFrmChangeKeyWords : public TForm
{
__published:
       TButton *OKBtn;
       TButton *CancelBtn;
       TBevel *Bevel1;
        TLabel *LblKeyWord;
        TEdit *EdKeyWord;
        TButton *BtnAdd;
        TListBox *LbxKeywords;
        TLabel *LblListBox;
        TButton *BtnDelete;
        TButton *BtnChange;
        TQuery *QryGeneral;
        void __fastcall EdKeyWordChange(TObject *Sender);
        void __fastcall BtnAddClick(TObject *Sender);
        void __fastcall LbxKeywordsClick(TObject *Sender);
        void __fastcall BtnChangeClick(TObject *Sender);
        void __fastcall BtnDeleteClick(TObject *Sender);
        void __fastcall OKBtnClick(TObject *Sender);
private:
     AnsiString SessionName;
     AnsiString DatabaseName;
     AnsiString ArticleID;
     bool KeyWords;
     AnsiString KeyFieldName;
     AnsiString TableName;
public:
       virtual __fastcall TFrmChangeKeyWords(TComponent* AOwner);
        void __fastcall SetUp(const AnsiString sessname,
                    const AnsiString dbname,AnsiString article_id,bool keywords);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmChangeKeyWords *FrmChangeKeyWords;
//---------------------------------------------------------------------------
#endif
```

DlgKeyWords.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "DlgKeyWords.h"
extern const int LengthKeyword=60;
extern const int LengthMathCode=5;
//---------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmChangeKeyWords *FrmChangeKeyWords;
//---------------------------------------------------------------------
__fastcall TFrmChangeKeyWords::TFrmChangeKeyWords(TComponent* AOwner)
      : TForm(AOwner)
{
}
//---------------------------------------------------------------------
void __fastcall TFrmChangeKeyWords::SetUp(const AnsiString sessname,
                const AnsiString dbname,AnsiString article_id,bool keywords){
   SessionName=sessname;
   DatabaseName=dbname;
   ArticleID=article_id;
   KeyWords= keywords;

   QryGeneral->DatabaseName=dbname;
   // get current list of keywords
   AnsiString Query;
   QryGeneral->Close();
   QryGeneral->SQL->Clear();
   if (KeyWords){
      KeyFieldName = "KeyWord";
      TableName = "ArticleKeyWords";
      EdKeyWord->MaxLength = LengthKeyword;
   } else {
      KeyFieldName = "MathCode";
      TableName = "ArticleMathCodes";
      EdKeyWord->MaxLength = LengthMathCode;
   }
   Query = "Select " + KeyFieldName + " from " + TableName +" where Article_ID ='";
   Query += ArticleID+ "'";
   QryGeneral->SQL->Add(Query);
   try {
      QryGeneral->Open();
      while(!QryGeneral->Eof) {
        LbxKeywords->Items->Add(
        QryGeneral->FieldByName(KeyFieldName)->AsString);
        QryGeneral->Next();
      }
   } catch (Exception& e) {
     ShowMessage(e.Message);
     // disable all buttons bar the leave (default state )
   }
   QryGeneral->Close();
}
//---------------------------------------------------------------------

void __fastcall TFrmChangeKeyWords::EdKeyWordChange(TObject *Sender)
{
   if (EdKeyWord->Text.Length() > 0)
     BtnAdd->Enabled=true;
   else
     BtnAdd->Enabled=false;
}
//---------------------------------------------------------------------
```

```
void __fastcall TFrmChangeKeyWords::BtnAddClick(TObject *Sender)
{
   int indx =  LbxKeywords->Items->IndexOf ( EdKeyWord->Text);
   if (indx >=0) {
     ShowMessage("Word is already in list");
     return;
   }
   LbxKeywords->Items->Add(EdKeyWord->Text);
   EdKeyWord->Text="";
   BtnAdd->Enabled =false;
   OKBtn->Enabled =true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmChangeKeyWords::LbxKeywordsClick(TObject *Sender)
{
  BtnChange->Enabled=true;
  BtnDelete->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmChangeKeyWords::BtnChangeClick(TObject *Sender)
{
   int indx = LbxKeywords->ItemIndex;
   EdKeyWord->Text = LbxKeywords->Items->Strings[indx];
   LbxKeywords->Items->Delete (indx);
   BtnChange->Enabled =false;
   BtnDelete->Enabled =false;
   OKBtn->Enabled =true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmChangeKeyWords::BtnDeleteClick(TObject *Sender)
{
   int indx = LbxKeywords->ItemIndex;
   LbxKeywords->Items->Delete (indx);
   BtnChange->Enabled =false;
   BtnDelete->Enabled =false;
   OKBtn->Enabled =true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmChangeKeyWords::OKBtnClick(TObject *Sender)
{
// this is where we update the keywords
   AnsiString Query;
    TSession* ts = Sessions->FindSession(SessionName);
    TDatabase* td = ts->FindDatabase(DatabaseName);
    td->StartTransaction();
    try {
       Query = "Delete from " + TableName + " where article_id='";
       Query += ArticleID +"'";
       QryGeneral->Close();
       QryGeneral->SQL->Clear();
       QryGeneral->SQL->Add(Query);
       QryGeneral->ExecSQL();
       QryGeneral->SQL->Clear();
       Query ="Insert into " + TableName + "(Article_ID," + KeyFieldName + ") ";
       Query += "VALUES (:Article_ID ,:" + KeyFieldName + " )";
       QryGeneral->SQL->Add(Query);
       QryGeneral->Prepare();
       QryGeneral->ParamByName("Article_ID")->AsString = ArticleID;
       for (int i=0; i<LbxKeywords->Items->Count;i++){
         AnsiString kw =  LbxKeywords->Items->Strings[i];
         QryGeneral->ParamByName(KeyFieldName)->AsString = kw;
         QryGeneral->ExecSQL();
       }
       QryGeneral->UnPrepare();
       td->Commit();
    } catch (Exception& e) {
      td->Rollback();
      ShowMessage(e.Message);
    }
}
//---------------------------------------------------------------------------
```

Module Name - DlgModArticles

DlgModArticles.h

```
//---------------------------------------------------------------------------
#ifndef DlgModArticlesH
#define DlgModArticlesH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
#include <DBCtrls.hpp>
#include <Mask.hpp>
#include <Menus.hpp>
#define ODBCVER 0x0300
#define ODBCVER 0x0300
#include <sqlext.h>
//---------------------------------------------------------------------------
class TDlgModifyArticle : public TForm
{
__published:
        TQuery *QryGeneral;
        TStoredProc *SProcModify;
        TStoredProc *SprocAdd;
        TPopupMenu *PmnuAbstract;
        TMenuItem *PmnuInsertImage;
        TMenuItem *PmnuChangeImage;
        TPanel *Panel1;
        TButton *OKBtn;
        TButton *CancelBtn;
        TPanel *Panel2;
        TMemo *MemoAbstract;
        TLabel *LblError;
        TLabel *Label7;
        TPanel *Panel3;
        TLabel *Label5;
        TMaskEdit *EdArticleId;
        TLabel *Label2;
        TEdit *EdReceiveDate;
        TLabel *Label3;
        TEdit *EdAcceptDate;
        TLabel *Label1;
        TEdit *EdArticleTitle;
        TLabel *Label4;
        TComboBox *CbxEditors;
        TRadioGroup *RgrpAvailable;
        TRadioGroup *RGrpPublished;
        TStoredProc *SprocArticleId;
        void __fastcall CancelBtnClick(TObject *Sender);
        void __fastcall OKBtnClick(TObject *Sender);
        void __fastcall CbxEditorsChange(TObject *Sender);
        void __fastcall EdAcceptDateExit(TObject *Sender);
        void __fastcall EdReceiveDateExit(TObject *Sender);
        void __fastcall PmnuAbstractPopup(TObject *Sender);
        void __fastcall PmnuInsertImageClick(TObject *Sender);
        void __fastcall PmnuChangeImageClick(TObject *Sender);
private:
        TDataSource *MyDSArticles;
        TMFullArticleObj* MyOrigData;

        bool AmModifying;
        bool Expanded;
        bool SettingUp;
        AnsiString DefaultEdStr;
        TStringList* MyImageList;
        char *ImStrt;
        char *ImEnd;
        int ImageStart;
        int ImageLen;
        int AltTxtStart;
        int AltTxtLen;
        int ImageMarkerLength;
        void __fastcall GetImageDetails(int act=0);
        SQLHANDLE hNativeDb;
        SQLHANDLE hstmt;
        bool __fastcall CursorInsideImageMarker();
        void __fastcall AddImageRecord(TMArticleImageObj* tio);
        void __fastcall UpdateAltText(TMArticleImageObj* tio);
        void __fastcall DeleteImageRecord(TMArticleImageObj* tio);
        void __fastcall CheckStatementRetResult(SQLRETURN retcode );
        void __fastcall WriteAnImageRecord(int filehandle,TMArticleImageObj* tio);
        void __fastcall UpdateAbstractField();
        void __fastcall WriteAbstractField();
        void __fastcall GetAbstractField();
        void __fastcall ReadAbstractField();


public:
       virtual __fastcall TDlgModifyArticle(TComponent* AOwner);
        virtual __fastcall ~TDlgModifyArticle();
        void SetUp(TDataSource* ads,const AnsiString &dbname,bool Modify);
};
//---------------------------------------------------------------------------
extern PACKAGE TDlgModifyArticle *DlgModifyArticle;
//---------------------------------------------------------------------------
#endif
```

DlgModArticles.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#include <vcl\clipbrd.hpp>
#pragma hdrstop
#include <fcntl.h>
#include <io.h>

#include "JipamCommon.h"
#include "DlgModArticles.h"
#include "DlgSelectImage.h"
#include "DMJipam.h"

#define MAXABSTRACTSIZE 6000


//---------------------------------------------------------------------
```

```
#pragma resource "*.dfm"
TDlgModifyArticle *DlgModifyArticle;
//---------------------------------------------------------------------------
__fastcall TDlgModifyArticle::TDlgModifyArticle(TComponent* AOwner)
        : TForm(AOwner)
{
   Expanded = false;
   MyImageList = new TStringList;
   MyOrigData = new TMFullArticleObj;
}
//---------------------------------------------------------------------------
__fastcall TDlgModifyArticle::~TDlgModifyArticle()
{
    for (int i=0; i< MyImageList->Count ; i++ ) {
     TMArticleImageObj* tob = (TMArticleImageObj*) MyImageList->Objects[i];
      delete tob;
   }
   delete MyImageList;
   for (int i=0; i < CbxEditors->Items->Count ; i++) {
      TMEditorObj* teob = (TMEditorObj *) CbxEditors->Items->Objects[i];
      delete teob;
   }
   delete MyOrigData;
}

//---------------------------------------------------------------------------
void
TDlgModifyArticle::SetUp(TDataSource* ads,
    const AnsiString & dbName,bool DoModify) {
   MyDSArticles=ads;
   QryGeneral->DatabaseName = dbName;
   SprocAdd->DatabaseName = dbName;
   SProcModify->DatabaseName =dbName;
   SprocArticleId->DatabaseName =dbName;
   AmModifying = DoModify;
   SettingUp=true;

   if (DoModify) { // we are modiying the underlying record
      RGrpPublished->Visible =true;
      MyOrigData->Article_id =
         MyDSArticles->DataSet->FieldByName("Article_Id")->AsString;
      MyOrigData->Title=
         MyDSArticles->DataSet->FieldByName("ArticleTitle")->AsString;
      MyOrigData->ReceiveDate =
         MyDSArticles->DataSet->FieldByName("ReceiveDate")->AsString;
      MyOrigData->AcceptDate =
         MyDSArticles->DataSet->FieldByName("AcceptDate")->AsString;
      MyOrigData->EditorId = MyDSArticles->DataSet->FieldByName("Editor_Id")-
>AsInteger;
      MyOrigData->Published =
              MyDSArticles->DataSet->FieldByName("Published")->AsBoolean;
      MyOrigData->Available =
              MyDSArticles->DataSet->FieldByName("Available")->AsBoolean;
      // retrieve the current abstract
      GetAbstractField();

   } else { // a new record
      SprocArticleId->ExecProc();
      MyOrigData->Article_id =
         SprocArticleId->Params->Items[1]->AsString ;
   }
   EdArticleId->Text    = MyOrigData->Article_id;
   EdArticleTitle->Text = MyOrigData->Title;
   EdReceiveDate->Text  = MyOrigData->ReceiveDate;
   EdAcceptDate->Text   = MyOrigData->AcceptDate;
   MemoAbstract->Text   = MyOrigData->Abstract;
   RGrpPublished->ItemIndex =1;
   RgrpAvailable->ItemIndex =1;
   if (MyOrigData->Published) { // a published article is in a volume not the place to
edit it
      RGrpPublished->ItemIndex =0;
      RgrpAvailable->ItemIndex =0;
      RgrpAvailable->Enabled = false;
   } else
       if (MyOrigData->Available) {
          RgrpAvailable->ItemIndex =0;
       }



   // fill the editor box
   QryGeneral->SQL->Clear();
   QryGeneral->SQL->Add("Select DisplayName,user_id from EditorList order by
Surname");
   QryGeneral->Open();
   DefaultEdStr = QryGeneral->FieldByName("DisplayName")->AsString;
   if (!DoModify) {
     MyOrigData->EditorId=QryGeneral->FieldByName("User_id")->AsInteger;
   }

   while (!QryGeneral->Eof) {
     TMEditorObj* teob = new TMEditorObj;
     teob->user_id = QryGeneral->FieldByName("User_id")->AsInteger;
     CbxEditors->Items->AddObject(
       QryGeneral->FieldByName("DisplayName")->AsString,teob);
     if (teob->user_id == MyOrigData->EditorId)
         DefaultEdStr= QryGeneral->FieldByName("DisplayName")->AsString;
     QryGeneral->Next();
   }
   QryGeneral->Close();

   CbxEditors->ItemIndex = CbxEditors->Items->IndexOf (DefaultEdStr);
// now get the artciles image records
   QryGeneral->SQL->Clear();
   AnsiString Query = "Select ImageNumber,ImageSource,ImageType,ImageAltText from
ArticleAbstractImages where article_id ='";
   Query += EdArticleId->Text + "'";
   QryGeneral->SQL->Add(Query);
   QryGeneral->Open();
   while (!QryGeneral->Eof) {
     TMArticleImageObj* tio = new TMArticleImageObj;
     tio->ImageNumber = QryGeneral->FieldByName("ImageNumber")->AsInteger;
     tio->SourceFileName = QryGeneral->FieldByName("ImageSource")->AsString;
     tio->SourceFileType = QryGeneral->FieldByName("ImageType")->AsString;
     tio->AltText = QryGeneral->FieldByName("ImageAltText")->AsString;
     AnsiString name = "image" + IntToStr(tio->ImageNumber);
     MyImageList->AddObject(name,tio);
     QryGeneral->Next();
   }
   QryGeneral->Close();


   OKBtn->Enabled=false;
   SettingUp=false;
}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgModArticles

```
void __fastcall TDlgModifyArticle::CancelBtnClick(TObject *Sender)
{
// do nothing but abandon changes
}
//---------------------------------------------------------------------------

void __fastcall TDlgModifyArticle::OKBtnClick(TObject *Sender)
{
// first pass through list check to see if images are found
   AnsiString ImMarker= "##image";
   AnsiString img;
   for (int k=0; k< MyImageList->Count ; k++) {
     img = "##";
        img += MyImageList->Strings[k] + "#";
     if ( MemoAbstract->Text.Pos(img) < 1) { // not found flag as deleted
        TMArticleImageObj * tio = (TMArticleImageObj *) MyImageList->Objects[k];
        tio->status = 3; // deleted
     }
   }
// second pass makes sure that all ##imagenn# strings have entries
   AnsiString LeftOver = MemoAbstract->Text;

   int off = LeftOver.Pos(ImMarker);
   while (off > 0) {
       int off1 = off+9;
       if (LeftOver.SubString(off1,1)== "#") img = LeftOver.SubString (off+2,7);
       else { off1--; img = LeftOver.SubString (off+2,6); }
       if (MyImageList->IndexOf (img) < 0) { // not found
         ShowMessage("Unable to find record for " + img);
         return;
       }
       LeftOver = LeftOver.SubString(off1+1,LeftOver.Length());
       off = LeftOver.Pos(ImMarker);
   }

//     DataModuleJipam->DbJipam->StartTransaction();
     try {
     TMEditorObj* teob = (TMEditorObj*)CbxEditors->Items->
                          Objects[CbxEditors->ItemIndex] ;

     if (AmModifying) {
     SProcModify->Params->Items[1]->AsString = EdArticleId->Text;
     SProcModify->Params->Items[2]->AsString = EdArticleTitle->Text;
     SProcModify->Params->Items[3]->AsString = EdReceiveDate->Text;
     SProcModify->Params->Items[4]->AsString = EdAcceptDate->Text;
     SProcModify->Params->Items[5]->AsInteger = teob->user_id;
     SProcModify->Params->Items[6]->AsString = ""; //MemoAbstract->Text;
     SProcModify->Params->Items[7]->AsString = RGrpPublished->Items->Strings
          [RGrpPublished->ItemIndex];
     SProcModify->Params->Items[8]->AsString = RgrpAvailable->Items->Strings
         [RgrpAvailable->ItemIndex];
     SProcModify->Prepare();
     SProcModify->ExecProc();

     if ( SProcModify->Params->Items[0]->AsInteger == 1) // should never happen
         ShowMessage("Article with ID " + EdArticleId->Text+ " No longer exists");

     } else { //add a record
     SprocAdd->Params->Items[1]->AsString = EdArticleId->Text;
     SprocAdd->Params->Items[2]->AsString = EdArticleTitle->Text;
     SprocAdd->Params->Items[3]->AsString = EdReceiveDate->Text;
     SprocAdd->Params->Items[4]->AsString = EdAcceptDate->Text;
     SprocAdd->Params->Items[5]->AsInteger = teob->user_id;
     SprocAdd->Params->Items[6]->AsString = ""; //MemoAbstract->Text;
     SprocAdd->Params->Items[7]->AsString = RGrpPublished->Items->Strings
         [RGrpPublished->ItemIndex];
     SprocAdd->Params->Items[8]->AsString = RgrpAvailable->Items->Strings
          [RgrpAvailable->ItemIndex];
     SprocAdd->Prepare();
     SprocAdd->ExecProc();
   }
   // because there is a size issue with parameters of 255 and because the
   // abstract is up to 6000 chars in size the following process is introduced
   UpdateAbstractField();
   // now for the image processing
   for (int k=0; k< MyImageList->Count ; k++) {
     TMArticleImageObj* tio = (TMArticleImageObj*)MyImageList->Objects[k] ;
     if (tio->status == 2) { // add record
       AddImageRecord(tio);
     } else { if (tio->status == 1) {
                 if (tio->AltTextChange) UpdateAltText(tio);
              } else { // records to delete
                 DeleteImageRecord(tio);
              }
     }

   }

   } catch (Exception & e) {
      ShowMessage(e.Message );
//      DataModuleJipam->DbJipam->Rollback();
     return;
   }
//   DataModuleJipam->DbJipam->Commit();
   ModalResult=mrOk;

}
//---------------------------------------------------------------------------



void __fastcall TDlgModifyArticle::CbxEditorsChange(TObject *Sender)
{
   if (SettingUp) return;
   OKBtn->Enabled = false;

    bool published =(RGrpPublished->Items->Strings
      [RGrpPublished->ItemIndex] == "Y" ? true : false);
   bool available = (RgrpAvailable->Items->Strings
      [RgrpAvailable->ItemIndex]== "Y" ? true : false);
   TMEditorObj* teob = (TMEditorObj*)CbxEditors->Items->Objects[CbxEditors->ItemIndex]
;
   if ((MyOrigData->Article_id != EdArticleId->Text ) ||
        (MyOrigData->Title      != EdArticleTitle->Text ) ||
        (MyOrigData->ReceiveDate != EdReceiveDate->Text ) ||
        (MyOrigData->AcceptDate  != EdAcceptDate->Text  ) ||
        (MyOrigData->EditorId    != teob->user_id ) ||
        (MyOrigData->Available   != available)            ||
        (MyOrigData->Published   != published)            ||
        (MyOrigData->Abstract    != MemoAbstract->Text ))
        OKBtn->Enabled = true;

}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgModArticles

```
void __fastcall TDlgModifyArticle::EdAcceptDateExit(TObject *Sender)
{
  LblError->Caption ="";
  OKBtn->Enabled = false;
  if (EdAcceptDate->Text.Trim().Length() > 1 ){
     try {
        TDateTime test(EdAcceptDate->Text);
     } catch (Exception& E) {
       LblError->Caption ="The AcceptDate date is not a valid date";
       // display message
       EdAcceptDate->SetFocus();
       return;
      }
  }
   OKBtn->Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TDlgModifyArticle::EdReceiveDateExit(TObject *Sender)
{
// check to see if valid date
   LblError->Caption ="";
   OKBtn->Enabled = false;
   if (EdReceiveDate->Text.Trim().Length() < 1 ) return;
   try {
      TDateTime test(EdReceiveDate->Text);
   } catch (Exception& E) {
     LblError->Caption ="The Receive date is not valid";
     // display message
     EdReceiveDate->SetFocus();
     return;
   }
   OKBtn->Enabled = true;
}
//---------------------------------------------------------------------------

bool __fastcall
TDlgModifyArticle::CursorInsideImageMarker(){

   char* ImMrker="##";
   char* TxtStrt = MemoAbstract->Text.c_str();
   char* Curpos = TxtStrt + MemoAbstract->SelStart;
// this is starts at 1; 0 means an empty string
   if (*Curpos==0) return false;
   char* AltTxt;
// a non empty abstract exists
   ImStrt = strstr(TxtStrt,ImMrker);
   while (ImStrt) {
// is the start less than this
     if (Curpos <= ImStrt)  // outside
        return false;
     TxtStrt = ImStrt+2; // skip start ##
     ImEnd = strstr(TxtStrt,ImMrker);
     if (ImEnd == 0) // invalid image pair marker
        return false;
// work out
     if (Curpos <= ImEnd+1) { // we have found it
        // find alt text stuff
        AltTxt = strchr(TxtStrt,'#');
        AltTxtStart=0;
        if (AltTxt < ImEnd){ //alt txt found
          AltTxtStart = AltTxt - MemoAbstract->Text.c_str()+2;
          AltTxtLen = ImEnd - AltTxt-1;
          ImageLen = AltTxt-TxtStrt;
        } else {
          ImageLen = ImEnd-TxtStrt;
        }
        ImageStart =  TxtStrt - MemoAbstract->Text.c_str()-1;
        ImageMarkerLength  = ImEnd - TxtStrt+4;
        return true;
     }
     TxtStrt = ImEnd+2;
     ImStrt = strstr(TxtStrt,ImMrker);
   }
   return false;
}

void __fastcall TDlgModifyArticle::PmnuAbstractPopup(TObject *Sender)
{
// check to see if we can enable the change image
   PmnuChangeImage->Enabled =false;
   PmnuInsertImage->Enabled = false;

   if (CursorInsideImageMarker())   PmnuChangeImage->Enabled =true;
   else            PmnuInsertImage->Enabled = true;
}
//---------------------------------------------------------------------------

void __fastcall TDlgModifyArticle::PmnuInsertImageClick(TObject *Sender)
{
// insert menu
   GetImageDetails(0);
}
//---------------------------------------------------------------------------

void __fastcall TDlgModifyArticle::PmnuChangeImageClick(TObject *Sender)
{
// change image

   GetImageDetails(1);
}
//---------------------------------------------------------------------------
// act=0 insert image
// act=1 change image
void __fastcall TDlgModifyArticle::GetImageDetails(int act){

   TStrings * tsl;
   int i;
   TMArticleImageObj* tob;
   TFrmSelectImage * Dlg;
   try {
     Dlg = new  TFrmSelectImage(this);
   } catch(Exception& e) { ShowMessage(e.Message); return; }
// before loading db with imagelist need to check to see if alttext has changed
   AnsiString image;
   AnsiString AltText;
   int itemno;
   if (act == 1) {
      image =MemoAbstract->Text.SubString(ImageStart+2,
         ImageLen );
      itemno = MyImageList->IndexOf (image);
      if (itemno >=0 && AltTxtStart ) { // this is item no we have on file
         // update the alt text from the memo box
         AltText = MemoAbstract->Text.SubString(AltTxtStart,AltTxtLen );
         tob = (TMArticleImageObj*) MyImageList ->Objects[itemno];
```

```
        if (tob->AltText != AltText) {
          tob->AltText = AltText;
          tob->AltTextChange =true;
        }
      }
   }

   Dlg->SetUp(MyImageList);
   if (act == 1) {
      int itemno = Dlg->LbxAvailable ->Items->IndexOf (image);
      Dlg->LbxAvailable->ItemIndex = itemno;
      Dlg->LbxAvailable->OnClick(this);
   }
   int rs =  Dlg->ShowModal();
   tsl=Dlg->LbxAvailable->Items;
   for (i=0; i< tsl->Count; i++) {
      tob = (TMArticleImageObj*) tsl->Objects[i];
      if (rs == mrOk) {
        if (tob->status == 4) {
          tob->status = 2;
          MyImageList->AddObject (tsl->Strings[i],tob);
        }
      } else {
        if (tob->status == 4) delete tob;
      }
   }
   if (rs == mrOk) {
      if (act > 0) {
        MemoAbstract->SelStart = ImageStart-1;
        MemoAbstract->SelLength = ImageMarkerLength;
      }
      tob = (TMArticleImageObj*) tsl->Objects[Dlg->LbxAvailable->ItemIndex];
      TClipboard* clp = Clipboard();
      AnsiString image = "##";
      image += tsl->Strings [Dlg->LbxAvailable->ItemIndex] + "#";
      image += tob->AltText + "##";
      clp->SetTextBuf(image.c_str());
      MemoAbstract->PasteFromClipboard ();
   }
   delete Dlg;
}
//---------------------------------------------------------------------------

//---------------------------------------------------------------------------
void
__fastcall TDlgModifyArticle::AddImageRecord(TMArticleImageObj* tio) {
// lets find out the ODBC connection handle

     unsigned short Size;
     if (DbiGetProp(DataModuleJipam->DbJipam->Handle, dbNATIVEHNDL,
                                  &hNativeDb, sizeof(hNativeDb), Size)
          != DBIERR_NONE)
        throw Exception("Cannot access ODBC Handle");
// now obtain the statement handle
     if (SQLAllocHandle(SQL_HANDLE_STMT, hNativeDb, &hstmt) != SQL_SUCCESS)
          throw Exception("Cannot allocate statement Handle");
     int FileHandle = open( tio->SourceFileName.c_str(),O_RDONLY | O_BINARY);
     if (FileHandle < 0) throw Exception(tio->SourceFileName + " is unabled to be
opened");
     TCursor temp = Screen->Cursor;
     Screen->Cursor = crHourGlass;
     try {
        WriteAnImageRecord(FileHandle,tio);
     } __finally {
        SQLFreeStmt(hstmt,SQL_CLOSE);
        close(FileHandle);
        Screen->Cursor = temp;
     }
}
//---------------------------------------------------------------------------
void __fastcall
TDlgModifyArticle::CheckStatementRetResult(SQLRETURN retcode ){
   SQLCHAR SqlState[6];
   SQLCHAR Msg[SQL_MAX_MESSAGE_LENGTH];
   SQLINTEGER NativeError;
   SQLSMALLINT MsgLen;

   if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO) ){
      if (retcode == SQL_INVALID_HANDLE)
         throw Exception("OBDC Invalid handle");
      // error
      SQLGetDiagRec(SQL_HANDLE_STMT,hstmt,1,SqlState,&NativeError,Msg,
          sizeof(Msg),&MsgLen);
      throw Exception((char*)Msg);
   }
}
//---------------------------------------------------------------------------

void
__fastcall TDlgModifyArticle::WriteAnImageRecord(int filehandle,
                                  TMArticleImageObj* tio) {
#define    MAX_DATA_LEN 4096
   char* Data[MAX_DATA_LEN];
   long FileLength = filelength(filehandle);

  SQLINTEGER   cbPDFFILE=0;
  SQLPOINTER   pToken;
  SQLRETURN    retcode;

  AnsiString Query ="INSERT INTO ArticleAbstractImages (ARTICLE_ID,ImageNumber," ;
  Query += "ImageSource,ImageType,ImageAltText,AnImage) values ('" + EdArticleId-
>Text;
  Query += "'," + IntToStr(tio->ImageNumber);
  Query += ",'" + tio->SourceFileName;
  Query += "','" + tio->SourceFileType + "','" + tio->AltText + "',?)";

  retcode = SQLPrepare(hstmt,Query.c_str(),SQL_NTS);
  if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO) )
      CheckStatementRetResult(retcode); // return failure

    /* Bind the parameters. For parameter 3, pass */
    /* the parameter number in ParameterValuePtr instead of a buffer */
    /* address.   */
    cbPDFFILE = SQL_LEN_DATA_AT_EXEC(FileLength);

    CheckStatementRetResult(SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT,
                    SQL_C_BINARY, SQL_LONGVARBINARY,
                    0, 0, (SQLPOINTER) 3, 0, &cbPDFFILE));
    /* Set values so data for parameter 3 will be */
    /* passed at execution. Note that the length parameter in */
    /* the macro SQL_LEN_DATA_AT_EXEC is 0. This assumes that */
    /* the driver returns "N" for the SQL_NEED_LONG_DATA_LEN */
    /* information type in SQLGetInfo.
                          */

    retcode = SQLExecute(hstmt);
    if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO
```

Appendix D – Module DlgModArticles

```
        || retcode == SQL_NEED_DATA) )
      CheckStatementRetResult(retcode); // return failure

    /* For data-at-execution parameters, call SQLParamData to */
    /* get the parameter number set by SQLBindParameter. */
    /* Call InitUserData. Call GetUserData and SQLPutData */
    /* repeatedly to get and put all data for the parameter. */
    /* Call SQLParamData to finish processing this parameter */

    while (retcode == SQL_NEED_DATA) {
        retcode = SQLParamData(hstmt, &pToken);
        if (retcode == SQL_NEED_DATA) {
            SQLINTEGER cbData,DataLeft;
            DataLeft = FileLength;

            while ( true) {
               cbData = DataLeft > MAX_DATA_LEN ? MAX_DATA_LEN : DataLeft;
               if (read(filehandle,Data,cbData) < 0)
                  throw Exception("Unable to read from file");
               SQLPutData(hstmt, Data, cbData);
               DataLeft -= cbData;
               if (DataLeft <= 0) break;
            }

        }
    }
    CheckStatementRetResult(retcode); // test for failure
    return; // flag that we were successful
}
//---------------------------------------------------------------------
void __fastcall
TDlgModifyArticle::UpdateAltText(TMArticleImageObj* tio){
   AnsiString Query = "Update ArticleAbstractImages set ImageAltText = '";
   Query += tio->AltText + "' where article_id ='";
   Query += EdArticleId->Text +  "' and ImageNumber = " + IntToStr(tio->ImageNumber);
   QryGeneral->Close();
   QryGeneral->SQL->Clear();
   QryGeneral->SQL->Add(Query);
   QryGeneral->ExecSQL ();
}
//---------------------------------------------------------------------
void __fastcall
TDlgModifyArticle::DeleteImageRecord(TMArticleImageObj* tio){
   AnsiString Query = "Delete from ArticleAbstractImages where article_id ='";
   Query += EdArticleId->Text +  "' and ImageNumber = " + IntToStr(tio->ImageNumber);
   QryGeneral->Close();
   QryGeneral->SQL->Clear();
   QryGeneral->SQL->Add(Query);
   QryGeneral->ExecSQL ();

}

//---------------------------------------------------------------------------
void
__fastcall TDlgModifyArticle::UpdateAbstractField() {
// lets find out the ODBC connection handle

     unsigned short Size;
     if (DbiGetProp(DataModuleJipam->DbJipam->Handle, dbNATIVEHNDL,
                                  &hNativeDb, sizeof(hNativeDb), Size)
          != DBIERR_NONE)
        throw Exception("Cannot access ODBC Handle");
// now obtain the statement handle
     if (SQLAllocHandle(SQL_HANDLE_STMT, hNativeDb, &hstmt) != SQL_SUCCESS)
          throw Exception("Cannot allocate statement Handle");
     TCursor temp = Screen->Cursor;
     Screen->Cursor = crHourGlass;
     try {
        WriteAbstractField();
     } __finally {
        SQLFreeStmt(hstmt,SQL_CLOSE);
        Screen->Cursor = temp;
     }
}

//---------------------------------------------------------------------

void
__fastcall TDlgModifyArticle::WriteAbstractField() {

#define    MAX_DATA_LEN 4096
  SQLINTEGER   cbPDFFILE=0;
  SQLPOINTER   pToken;
  SQLRETURN    retcode;

   QryGeneral->Close();
   AnsiString Query = "Update articles set abstract= ? where article_id = '";
   Query += EdArticleId->Text + "'";

  retcode = SQLPrepare(hstmt,Query.c_str(),SQL_NTS);
  if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO) )
      CheckStatementRetResult(retcode); // return failure

    /* Bind the parameters. For parameter 3, pass */
    /* the parameter number in ParameterValuePtr instead of a buffer */
    /* address.   */
    cbPDFFILE = SQL_LEN_DATA_AT_EXEC(MemoAbstract->Text.Length());

    CheckStatementRetResult(SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT,
                    SQL_C_CHAR, SQL_LONGVARCHAR,
                    6000, 0, (SQLPOINTER) 3, 0, &cbPDFFILE));
    /* Set values so data for parameter 3 will be */
    /* passed at execution. Note that the length parameter in */
    /* the macro SQL_LEN_DATA_AT_EXEC is 0. This assumes that */
    /* the driver returns "N" for the SQL_NEED_LONG_DATA_LEN */
    /* information type in SQLGetInfo.
                              */

    retcode = SQLExecute(hstmt);
    if (!(retcode == SQL_SUCCESS || retcode == SQL_SUCCESS_WITH_INFO
         || retcode == SQL_NEED_DATA) )
      CheckStatementRetResult(retcode); // return failure

    /* For data-at-execution parameters, call SQLParamData to */
    /* get the parameter number set by SQLBindParameter. */
    /* Call InitUserData. Call GetUserData and SQLPutData */
    /* repeatedly to get and put all data for the parameter. */
    /* Call SQLParamData to finish processing this parameter */

    while (retcode == SQL_NEED_DATA) {
        retcode = SQLParamData(hstmt, &pToken);
        if (retcode == SQL_NEED_DATA) {
            SQLINTEGER cbData,DataLeft;
            DataLeft = MemoAbstract->Text.Length();
            char* Data = MemoAbstract->Text.c_str();
            while ( true) {
               cbData = DataLeft > MAX_DATA_LEN ? MAX_DATA_LEN : DataLeft;
```

```
            retcode = SQLPutData(hstmt, Data, cbData);
            DataLeft -= cbData;
            Data += cbData;
            if (DataLeft <= 0) break;
          }
          retcode = SQL_NEED_DATA;

        }
    }
    CheckStatementRetResult(retcode); // test for failure
    return; // flag that we were successful
}

//---------------------------------------------------------------------------
void
__fastcall TDlgModifyArticle::GetAbstractField() {
// lets find out the ODBC connection handle

     unsigned short Size;
     if (DbiGetProp(DataModuleJipam->DbJipam->Handle, dbNATIVEHNDL,
                                    &hNativeDb, sizeof(hNativeDb), Size)
         != DBIERR_NONE)
       throw Exception("Cannot access ODBC Handle");
// now obtain the statement handle
     if (SQLAllocHandle(SQL_HANDLE_STMT, hNativeDb, &hstmt) != SQL_SUCCESS)
          throw Exception("Cannot allocate statement Handle");
     TCursor temp = Screen->Cursor;
     Screen->Cursor = crHourGlass;
     try {
        ReadAbstractField();
     } __finally {
        SQLFreeStmt(hstmt,SQL_CLOSE);
        Screen->Cursor = temp;
     }
}
//---------------------------------------------------------------------

void
__fastcall TDlgModifyArticle::ReadAbstractField() {
  char DataIn[MAXABSTRACTSIZE];
  SQLINTEGER  sDataIn;
  SQLRETURN   retcode;
AnsiString Query="Select cast(abstract as text) from articles where article_id = '";
  Query+= MyOrigData->Article_id + "'";
  retcode = SQLExecDirect(hstmt, Query.c_str(), SQL_NTS);
  CheckStatementRetResult(retcode); // test for failure
retcode = SQLFetch(hstmt);
  CheckStatementRetResult(retcode); // test for failure
  retcode = SQLGetData(hstmt, 1, SQL_C_DEFAULT,DataIn, MAXABSTRACTSIZE, &sDataIn);
  CheckStatementRetResult(retcode); // test for failure
  MyOrigData->Abstract = DataIn;
  return; // flag that we were successful
}
```

Appendix D – Module DlgModArticles

Module Name - DlgModIntitution

DlgModIntitution.h

```
//---------------------------------------------------------------------------
#ifndef DlgModInstitutionH
#define DlgModInstitutionH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
//---------------------------------------------------------------------------
class TFrmModInstitution : public TForm
{
__published:
        TButton *CancelBtn;
        TBevel *Bevel1;
          TLabel *Label1;
          TLabel *Label2;
          TLabel *Label3;
          TLabel *Label4;
          TLabel *Label5;
          TLabel *Label6;
          TLabel *Label7;
          TLabel *Label8;
          TButton *BtnUpdate;
          TEdit *EdSmallDesc;
          TEdit *EdAddress1;
          TEdit *EdAddress2;
          TEdit *EdAddress3;
          TEdit *EdAddress4;
          TEdit *EdAddress5;
          TEdit *EdTelephone;
          TEdit *EdFascimile;
          TLabel *LblError;
          TStoredProc *SprocAdd;
          TStoredProc *SprocModify;
          void __fastcall EdSmallDescExit(TObject *Sender);
          void __fastcall BtnUpdateClick(TObject *Sender);
          void __fastcall EdSmallDescChange(TObject *Sender);
private:
          AnsiString DatabaseName;
          bool AmModifying;
          int Institution_id;
public:
        virtual __fastcall TFrmModInstitution(TComponent* AOwner);
          void __fastcall SetUp(const AnsiString dbname,bool ammodifying,
            int institutionid=0);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmModInstitution *FrmModInstitution;
//---------------------------------------------------------------------------
#endif
```

DlgModIntitution.Cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include "JipamCommon.h"

#include "DlgModInstitution.h"
extern const int LengthSmallDesc=60;
extern const int LengthAddressLine1=255;
extern const int LengthAddressLine2=255;
extern const int LengthAddressLine3=255;
extern const int LengthAddressLine4=255;
extern const int LengthAddressLine5=255;
extern const int LengthTelephone=60;
extern const int LengthFascimile=60;
//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmModInstitution *FrmModInstitution;
//---------------------------------------------------------------------------
__fastcall TFrmModInstitution::TFrmModInstitution(TComponent* AOwner)
        : TForm(AOwner)
{
  AmModifying=false;
}
//---------------------------------------------------------------------------
void __fastcall TFrmModInstitution::SetUp(const AnsiString dbname,bool ammodifying,
          int institutionid){
   DatabaseName = dbname;
   AmModifying=  ammodifying;
   Institution_id= institutionid;
   SprocAdd->DatabaseName =dbname;
   SprocModify->DatabaseName =dbname;
   EdSmallDesc->MaxLength = LengthSmallDesc;
   EdAddress1->MaxLength = LengthAddressLine1;
   EdAddress2->MaxLength = LengthAddressLine2;
   EdAddress3->MaxLength = LengthAddressLine3;
   EdAddress4->MaxLength = LengthAddressLine4;
   EdAddress5->MaxLength = LengthAddressLine5;
   EdTelephone->MaxLength = LengthTelephone;
   EdFascimile->MaxLength = LengthFascimile;
   if (AmModifying) {
      BtnUpdate->Caption="Update";
      Caption="Modifying an institution";
   }
   BtnUpdate->Enabled =false;

}
//---------------------------------------------------------------------------

void __fastcall TFrmModInstitution::EdSmallDescExit(TObject *Sender)
{
  LblError->Caption = "";
  if ( EdSmallDesc->Text.Trim().Length() < 1)
    BtnUpdate->Enabled=false;
  else
    BtnUpdate->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmModInstitution::BtnUpdateClick(TObject *Sender)
{
```

```
// when update is clicked
   try {
   if (AmModifying) {
   SprocModify->Params->Items[1]->AsString = EdSmallDesc->Text;
   SprocModify->Params->Items[2]->AsString = EdAddress1->Text;
   SprocModify->Params->Items[3]->AsString = EdAddress2->Text;
   SprocModify->Params->Items[4]->AsString = EdAddress3->Text;
   SprocModify->Params->Items[5]->AsString = EdAddress4->Text;
   SprocModify->Params->Items[6]->AsString = EdAddress5->Text;
   SprocModify->Params->Items[7]->AsString = EdTelephone->Text;
   SprocModify->Params->Items[8]->AsString = EdFascimile->Text;
   SprocModify->Params->Items[9]->AsInteger = Institution_id;
   SprocModify->Prepare();
   SprocModify->ExecProc();
   int res =SprocModify->Params->Items[0]->AsInteger;
     if (res  < 0 ) {
       LblError->Caption = "Small Desc already exists in table";
       EdSmallDesc->SetFocus();
       return;
     }
     if (res == 0){ // should never happen
       LblError->Caption = "The institution was deleted else where";
       EdSmallDesc->SetFocus();
       return;
     }
   } else { //add a record
   SprocAdd->Params->Items[1]->AsString = EdSmallDesc->Text;
   SprocAdd->Params->Items[2]->AsString = EdAddress1->Text;
   SprocAdd->Params->Items[3]->AsString = EdAddress2->Text;
   SprocAdd->Params->Items[4]->AsString = EdAddress3->Text;
   SprocAdd->Params->Items[5]->AsString = EdAddress4->Text;
   SprocAdd->Params->Items[6]->AsString = EdAddress5->Text;
   SprocAdd->Params->Items[7]->AsString = EdTelephone->Text;
   SprocAdd->Params->Items[8]->AsString = EdFascimile->Text;
   SprocAdd->Prepare();
   SprocAdd->ExecProc();
   if ( SprocAdd->Params->Items[0]->AsInteger < 1) {
     LblError->Caption = "Small Desc already exists in table";
     EdSmallDesc->SetFocus();
     return;
   }
   }
   ModalResult=mrOk;
   return;
   } catch(Exception& e) {
     ShowMessage(e.Message);
   }
}
//---------------------------------------------------------------------------

void __fastcall TFrmModInstitution::EdSmallDescChange(TObject *Sender)
{
  BtnUpdate->Enabled=true;
}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgModIntitution

MODULE NAME - DLGNEWVOLUME

DLGNEWVOLUME.H

```
//---------------------------------------------------------------------------
#ifndef DlgNewVolumeH
#define DlgNewVolumeH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
#include <Mask.hpp>
//---------------------------------------------------------------------------
class TFrmNewVolume : public TForm
{
__published:
        TButton *CancelBtn;
        TBevel *Bevel1;
         TLabel *Label1;
         TLabel *Label2;
         TLabel *Label3;
         TLabel *Label4;
         TEdit *EdIssueDate;
         TQuery *QryGeneral;
         TStoredProc *SprocAdd;
         TStoredProc *SprocMod;
         TButton *BtnUpdate;
         TMaskEdit *EdVolumeNo;
         TMaskEdit *EdYear;
         TMaskEdit *EdIssueNo;
         void __fastcall EdVolumeNoChange(TObject *Sender);
         void __fastcall EdIssueDateExit(TObject *Sender);
         void __fastcall OKBtnClick(TObject *Sender);
private:
         AnsiString DatabaseName;
         TMVolObject* TMvb;
         bool AmModifying;
public:
        virtual __fastcall TFrmNewVolume(TComponent* AOwner);
         void __fastcall SetUp(const AnsiString dbname,TMVolObject* tob,bool
ammodiying);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmNewVolume *FrmNewVolume;
//---------------------------------------------------------------------------
#endif
```

DLGNEWVOLUME.CPP

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include "Jipamcommon.h"
#include "DlgNewVolume.h"
//---------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmNewVolume *FrmNewVolume;
//---------------------------------------------------------------------
__fastcall TFrmNewVolume::TFrmNewVolume(TComponent* AOwner)
        : TForm(AOwner)
{
  AmModifying=false;
}
//---------------------------------------------------------------------
void __fastcall TFrmNewVolume::SetUp(const AnsiString dbname,TMVolObject* tvb,
          bool ammodiying){
   DatabaseName=dbname;
   TMvb=tvb;
   AmModifying=ammodiying;
   EdVolumeNo->Text = IntToStr(tvb->VOLUME_NO);
   EdIssueNo->Text = IntToStr(tvb->ISSUE_NO);
   EdYear->Text = IntToStr(tvb->ISSUEYear);
   EdIssueDate->Text = tvb->DateOfIssue.FormatString("dd/mm/yy");
   BtnUpdate->Enabled=false;
}
//---------------------------------------------------------------------

void __fastcall TFrmNewVolume::EdVolumeNoChange(TObject *Sender)
{
  BtnUpdate->Enabled=true;
}
//---------------------------------------------------------------------


void __fastcall TFrmNewVolume::EdIssueDateExit(TObject *Sender)
{
   try {
     TDateTime v(EdIssueDate->Text);
   } catch (Exception& e) {
     ShowMessage(e.Message);
     EdIssueDate->SetFocus();
   }
}
//---------------------------------------------------------------------

void __fastcall TFrmNewVolume::OKBtnClick(TObject *Sender)
{
   try {
   if (AmModifying) {
     SprocMod->Params->Items[1]->AsInteger = StrToInt(EdVolumeNo->Text.Trim());
     SprocMod->Params->Items[2]->AsInteger = StrToInt(EdIssueNo->Text.Trim());
     SprocMod->Params->Items[3]->AsInteger = StrToInt(EdYear->Text.Trim());
     SprocMod->Params->Items[4]->AsString = EdIssueDate->Text.Trim();
     SprocMod->Params->Items[5]->AsInteger = TMvb->Volume_id;
     SprocMod->Prepare();
     SprocMod->ExecProc();

     if ( SprocMod->Params->Items[0]->AsInteger < 1) { // should never happen
       ShowMessage("Volume with ID " + IntToStr(TMvb->Volume_id)+ " No longer
exists");
       return;
     }
   } else { //add a record
     SprocAdd->Params->Items[1]->AsInteger = StrToInt(EdVolumeNo->Text.Trim());
```

```
      SprocAdd->Params->Items[2]->AsInteger = StrToInt(EdIssueNo->Text.Trim());
      SprocAdd->Params->Items[3]->AsInteger = StrToInt(EdYear->Text.Trim());
      SprocAdd->Params->Items[4]->AsString = EdIssueDate->Text.Trim();
      SprocAdd->Prepare();
      SprocAdd->ExecProc();
      int   vol_id = SprocAdd->Params->Items[0]->AsInteger;
      if ( vol_id < 1) {// should never happen
        ShowMessage("Volume already exists - not added");
        return;
      }
    }
      ModalResult=mrOk;
      return;
    } catch (Exception& e) {
      ShowMessage(e.Message);
      return;
    }
}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgNewVolume

Module Name – DlgSelectImage

DlgSelectImage.h

```
//---------------------------------------------------------------------------
#ifndef DlgSelectImageH
#define DlgSelectImageH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Dialogs.hpp>
//---------------------------------------------------------------------------
class TFrmSelectImage : public TForm
{
__published:
        TButton *OKBtn;
         TButton *Close;
        TBevel *Bevell;
         TListBox *LbxAvailable;
         TLabel *Labell;
         TLabel *LblFilename;
         TLabel *Label3;
         TButton *BtnAdd;
         TOpenDialog *DlgNewImageSelection;
         TLabel *Label2;
         TEdit *EdAltText;
         void __fastcall LbxAvailableClick(TObject *Sender);
         void __fastcall BtnAddClick(TObject *Sender);
         void __fastcall EdAltTextExit(TObject *Sender);
private:
     int NextValue;
public:
        virtual __fastcall TFrmSelectImage(TComponent* AOwner);
        virtual __fastcall ~TFrmSelectImage();
         void __fastcall SetUp(TStringList* tsl);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmSelectImage *FrmSelectImage;
//---------------------------------------------------------------------------
#endif
```

DlgSelectImage.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "DlgSelectImage.h"
#include "JipamCommon.h"

//---------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmSelectImage *FrmSelectImage;
//---------------------------------------------------------------------
__fastcall TFrmSelectImage::TFrmSelectImage(TComponent* AOwner)
        : TForm(AOwner)
{
   NextValue=1;
}
//---------------------------------------------------------------------
__fastcall TFrmSelectImage::~TFrmSelectImage(){

}

//---------------------------------------------------------------------
void __fastcall TFrmSelectImage::SetUp(TStringList* tsl){
  LbxAvailable->Items->Assign(tsl);
  NextValue=1;
  if (LbxAvailable->Items->Count > 0 ) { //we have existing list
    for (int i=0; i< LbxAvailable->Items->Count; i++) {
       TMArticleImageObj* tob = (TMArticleImageObj*) LbxAvailable->Items->Objects[i];
       if (tob->ImageNumber >= NextValue) NextValue = tob->ImageNumber+1;
    }
  }
}
//---------------------------------------------------------------------
void __fastcall TFrmSelectImage::LbxAvailableClick(TObject *Sender)
{
// a single click takes the current
    OKBtn->Enabled=false;
    EdAltText->Enabled =false;
    EdAltText->Text ="";
    LblFilename->Caption ="";
    if (LbxAvailable->ItemIndex < 0) return;
    TMArticleImageObj* tob = (TMArticleImageObj*) LbxAvailable->Items->Objects[
        LbxAvailable->ItemIndex];
    LblFilename->Caption =  tob->SourceFileName;
    EdAltText->Text = tob->AltText;
    EdAltText->Enabled = true;
    if (EdAltText->Text.Trim().Length () > 0)    OKBtn->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmSelectImage::BtnAddClick(TObject *Sender)
{
// add another image
  if (DlgNewImageSelection->Execute()) {
     TMArticleImageObj* tob = new TMArticleImageObj;
     tob->status = 4; //new NB converted to 2 on successful return
     tob->ImageNumber = NextValue++  ; // image number
     tob->SourceFileName =  DlgNewImageSelection->FileName;
     tob->SourceFileType =  tob->SourceFileName.SubString(tob-
>SourceFileName.Length()-2,3);
     AnsiString name = "image";
     name+= IntToStr(tob->ImageNumber);
     tob->AltText = name;
     tob->AltTextChange = true;
     EdAltText->Text=name;
     LbxAvailable->Items->AddObject (name,tob);
     LbxAvailable->ItemIndex = LbxAvailable->Items->IndexOf(name);
     LbxAvailable->OnClick(this);
  }
}
```

```
//---------------------------------------------------------------------------


void __fastcall TFrmSelectImage::EdAltTextExit(TObject *Sender)
{
  TMArticleImageObj* tob = (TMArticleImageObj*) LbxAvailable->Items->Objects[
        LbxAvailable->ItemIndex];
  if (tob->AltText != EdAltText->Text ) {
     tob->AltText = EdAltText->Text;
     tob->AltTextChange = true;
  }
}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgSelectImage

Module Name - DlgSelectNewAuthor

DlgSelectNewAuthor.h

```
//---------------------------------------------------------------------------
#ifndef DlgSelectNewAuthorH
#define DlgSelectNewAuthorH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
//---------------------------------------------------------------------------
class TFrmSelectNewAuthor : public TForm
{
__published:
        TButton *OKBtn;
        TButton *CancelBtn;
        TBevel *Bevel1;
         TEdit *EdFilterText;
         TLabel *Label1;
         TListBox *LbxNewAuthors;
         TButton *BtnGet;
         TQuery *QryGeneral;
         void __fastcall BtnGetClick(TObject *Sender);
         void __fastcall EdFilterTextChange(TObject *Sender);
         void __fastcall LbxNewAuthorsClick(TObject *Sender);
         void __fastcall OKBtnClick(TObject *Sender);
private:
         AnsiString DatabaseName;
         void __fastcall ClearObjects(TListBox* tlb);
public:
        virtual __fastcall TFrmSelectNewAuthor(TComponent* AOwner);
         virtual __fastcall ~TFrmSelectNewAuthor();
         void __fastcall SetUp(const AnsiString dbname);
         bool Successful;
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmSelectNewAuthor *FrmSelectNewAuthor;
//---------------------------------------------------------------------------
#endif
```

DlgSelectNewAuthor.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "DlgSelectNewAuthor.h"
#include "JipamCommon.h"
//---------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmSelectNewAuthor *FrmSelectNewAuthor;
//---------------------------------------------------------------------
__fastcall TFrmSelectNewAuthor::TFrmSelectNewAuthor(TComponent* AOwner)
        : TForm(AOwner)
{
}
__fastcall TFrmSelectNewAuthor::~TFrmSelectNewAuthor(){
  ClearObjects(LbxNewAuthors);
}
//---------------------------------------------------------------------
void __fastcall TFrmSelectNewAuthor::SetUp(const AnsiString dbname){
  DatabaseName= dbname;
  QryGeneral->DatabaseName=dbname;
  Successful=false;

}
//---------------------------------------------------------------------

void __fastcall TFrmSelectNewAuthor::BtnGetClick(TObject *Sender)
{
//
   AnsiString Query;
   Query="Select user_id,Surname,FirstName,OtherInits,Institution_id,UserRole from
Users where Surname like '";
   Query+= EdFilterText->Text.Trim();
   Query+= "%' and UserRole in (0,3)";
   QryGeneral->Close();

   ClearObjects(LbxNewAuthors);
   LbxNewAuthors->Clear();
   OKBtn->Enabled=false;

   QryGeneral->SQL->Clear();
   QryGeneral->SQL->Add(Query);
   try {
     QryGeneral->Open();
     while (! QryGeneral->Eof) {
         AnsiString Auth = QryGeneral->FieldByName("Surname")->AsString;
         Auth+= "," +  QryGeneral->FieldByName("FirstName")->AsString;
         Auth+= " " +  QryGeneral->FieldByName("OtherInits")->AsString;
         TMObject* tob = new TMObject(
            QryGeneral->FieldByName("User_id")->AsInteger,
            QryGeneral->FieldByName("Institution_id")->AsInteger,
            QryGeneral->FieldByName("UserRole")->AsInteger);
         LbxNewAuthors->Items->AddObject(Auth,tob);

        QryGeneral->Next();
     }
     QryGeneral->Close();
   } catch (Exception& e) {
     LbxNewAuthors->Enabled=false;
   }
   BtnGet->Enabled=false;
}
//---------------------------------------------------------------------

void __fastcall TFrmSelectNewAuthor::EdFilterTextChange(TObject *Sender)
{
//
   if (EdFilterText->Text.Trim().Length() > 0) BtnGet->Enabled = true;
   else                                        BtnGet->Enabled = false;
}
```

```
//---------------------------------------------------------------------------
void __fastcall TFrmSelectNewAuthor::ClearObjects(TListBox* tlb){
  for (int i=0; i<tlb->Items->Count; i++){
     TMObject* tob = (TMObject*) tlb->Items->Objects[i];
     delete tob;
  }
}
//---------------------------------------------------------------------------

void __fastcall TFrmSelectNewAuthor::LbxNewAuthorsClick(TObject *Sender)
{
// select an item in the listbox
   OKBtn->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmSelectNewAuthor::OKBtnClick(TObject *Sender)
{
// we have chosen an item
// As the listbox entries are userroles 0 or 3 we need to update their role
// in users to reflect their new status
// the entry is pass back to the calling process
    AnsiString Query;
   QryGeneral->Close();
   QryGeneral->SQL->Clear();
   TMObject* tob = (TMObject*) LbxNewAuthors->Items->Objects[
      LbxNewAuthors->ItemIndex ];
   Query= "Update users set UserRole= " ;
   if (tob->GetUserRole() == 3) { // Editor becoming author as well
     Query+= "4";
   } else { // only a user
     Query+= "2";
   }
   Query+= " where user_id = " + IntToStr(tob->GetUser_id());
   QryGeneral->SQL->Add(Query);
   try {
     QryGeneral->ExecSQL();
     Successful=true;
   } catch (Exception& e) {
     ShowMessage(e.Message);
   }

}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgSelectNewAuthor

Module Name – DlgUserModify

DlgUserModify.h

```
//---------------------------------------------------------------------------
#ifndef DlgUserModifyH
#define DlgUserModifyH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
#include <Mask.hpp>
//---------------------------------------------------------------------------
class TDlgModifyUser : public TForm
{
__published:
        TButton *BtnUpdate;
       TButton *CancelBtn;
       TBevel *Bevel1;
        TLabel *Label1;
        TLabel *Label2;
        TLabel *Label3;
        TLabel *Label4;
        TLabel *Label5;
        TLabel *Label6;
        TEdit *EdStartDate;
        TEdit *EdRenewalDate;
        TButton *BtnNext;
        TButton *BtnPrior;
        TLabel *LblError;
        TQuery *QryModUser;
        TLabel *Label7;
        TMaskEdit *EdOtherInits;
        TMaskEdit *EdSurname;
        TLabel *Label8;
        TLabel *Label9;
        TMaskEdit *EdAffliation;
        TLabel *Label10;
        TMaskEdit *EdDisplayName;
        TLabel *Label11;
        TComboBox *CbxInstitution;
        TComboBox *CbxUserRole;
        TLabel *Label12;
        TLabel *Label13;
        TLabel *Label14;
        TMaskEdit *EdEmailAddress;
        TMaskEdit *EdPassword;
        TMaskEdit *EdFirstName;
        TMaskEdit *EdWebAddress;
        void __fastcall BtnUpdateClick(TObject *Sender);
        void __fastcall BtnNextClick(TObject *Sender);
        void __fastcall BtnPriorClick(TObject *Sender);
        void __fastcall EdRenewalDateExit(TObject *Sender);
        void __fastcall CancelBtnClick(TObject *Sender);
        void __fastcall CbxUserRoleExit(TObject *Sender);
private:
        UserRecord* Ur;
        bool    SetUpInProgress;
        bool    DbNotFailed;
        void __fastcall SetUpdateButton();
public:
       virtual __fastcall TDlgModifyUser(TComponent* AOwner);
        int Direction; // 1= prior 2 = next;
        void SetUpRecord();
        void SetUp(UserRecord* ur,const AnsiString& dbname);
};
//---------------------------------------------------------------------------
extern PACKAGE TDlgModifyUser *DlgModifyUser;
//---------------------------------------------------------------------------
#endif
```

DlgUserModify.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgUserModify.h"
extern char* EDMPassword;
extern const int LengthEmailaddress;
extern const int LengthPassword;
extern const int LengthFirstName;
extern const int LengthOtherInits;
extern const int LengthSurname;
extern const int LengthDisplayName;
extern const int LengthWebAddress;
extern const int LengthAffliation;

//---------------------------------------------------------------------
#pragma resource "*.dfm"
TDlgModifyUser *DlgModifyUser;
//---------------------------------------------------------------------
__fastcall TDlgModifyUser::TDlgModifyUser(TComponent* AOwner)
      : TForm(AOwner)
{
  Direction=2;
}
//---------------------------------------------------------------------
void __fastcall TDlgModifyUser::BtnUpdateClick(TObject *Sender)
{
// See if we can update the current record
   bool CommaRequired=false;
   AnsiString Query = "SET DATEFORMAT dmy Update Users Set ";
   if (EdEmailAddress->Text != Ur->EmailAddress) {
       Query += "EmailAddress = '"+EdEmailAddress->Text.Trim()+"' ";
       CommaRequired =true;
   }
   if (EdPassword->Text.Trim() != Ur->UserPassword) {
       if (CommaRequired) Query+=",";
       Query += "UserPassword = '"+EdPassword->Text.Trim()+"' ";
       CommaRequired =true;
   }
   if (EdFirstName->Text != Ur->FirstName) {
       if (CommaRequired) Query+=",";
       Query += "FirstName = '"+EdFirstName->Text.Trim()+"' ";
```

```
    CommaRequired =true;
}
if (EdOtherInits->Text != Ur->OtherInits) {
    if (CommaRequired) Query+=",";
    Query += "OtherInits = '"+EdOtherInits->Text.Trim()+"' ";
    CommaRequired =true;
}
if (EdSurname->Text != Ur->Surname) {
    if (CommaRequired) Query+=",";
    Query += "Surname = '"+EdSurname->Text.Trim()+"' ";
    CommaRequired =true;
}
if (EdWebAddress->Text != Ur->WebAddress) {
    if (CommaRequired) Query+=",";
    Query += "WebAddress = '"+EdWebAddress->Text.Trim()+"' ";
    CommaRequired =true;
}
if (EdAffliation->Text != Ur->Affiliation) {
    if (CommaRequired) Query+=",";
    Query += "Affiliation = '"+EdAffliation->Text.Trim()+"' ";
    CommaRequired =true;
}

if (EdDisplayName->Text != Ur->DisplayName) {
    if (CommaRequired) Query+=",";
    Query += "DisplayName = '"+EdDisplayName->Text.Trim()+"' ";
    CommaRequired =true;
}
if ( CbxInstitution->ItemIndex!=Ur->ItemIndex) {
  QryModUser->SQL->Clear();
  AnsiString Queryl = "Select Institution_ID from Institutions where SmallDesc='";
  Queryl += CbxInstitution->Items->Strings[CbxInstitution->ItemIndex] + "'";
  QryModUser->SQL->Add(Queryl);
  QryModUser->Open();
  int instiD=QryModUser->FieldByName("Institution_ID")->AsInteger;
  QryModUser->Close();
  if (CommaRequired) Query+=",";
  Query += "Institution_ID = "+IntToStr(instiD)+" ";
  CommaRequired =true;

};
if (CbxUserRole->ItemIndex != Ur->UserRole) {
   if (CommaRequired) Query+=",";
   Query += "UserRole = "+ IntToStr(CbxUserRole->ItemIndex);
   Query += " ";
   CommaRequired =true;
}

 if (EdStartDate->Text != Ur->StartDate.DateString()) {
     if (CommaRequired) Query+=",";
     Query += "StartDate = '"+EdStartDate->Text+"' ";
     CommaRequired =true;
 }
 if (EdRenewalDate->Text != Ur->RenewalDate.DateString()) {
     if (CommaRequired) Query+=",";
     Query += "RenewalDate = '"+EdRenewalDate->Text+"' ";
 }
 Query+=" where User_id = "+ IntToStr(Ur->User_id);
 QryModUser->SQL->Clear();
 QryModUser->SQL->Add(Query);
 try {
   QryModUser->ExecSQL();
   LblError->Caption = EdEmailAddress->Text + " Updated successfully";
 } catch ( Exception& er) {
    ShowMessage(er.Message);
      LblError->Caption = EdEmailAddress->Text + " Update failed";
      DbNotFailed=false;
      BtnPrior->Enabled=false;
      BtnNext->Enabled=false;
   }
   BtnUpdate->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TDlgModifyUser::BtnNextClick(TObject *Sender)
{
   if (BtnUpdate->Enabled) {
      if (MessageDlg("Update Record?", mtConfirmation,
               TMsgDlgButtons() << mbYes << mbNo, 0) == mrYes) {
       BtnUpdateClick(Sender);
       if (BtnUpdate->Enabled) return; // failed
       }
   }
// go to next record if we can
   LblError->Caption="";
   Direction=2;
   ModalResult=mrYes;
}
//---------------------------------------------------------------------------

void __fastcall TDlgModifyUser::BtnPriorClick(TObject *Sender)
{
   if (BtnUpdate->Enabled) {
      if (MessageDlg("Update Record?", mtConfirmation,
               TMsgDlgButtons() << mbYes << mbNo, 0) == mrYes) {
       BtnUpdateClick(Sender);
       if (BtnUpdate->Enabled) return; // failed
       }
   }
// go to prior record if we can
   LblError->Caption="";
   Direction=1;
   ModalResult=mrYes;
}
//---------------------------------------------------------------------------
void
TDlgModifyUser::SetUpRecord(){
   SetUpInProgress=true;

   EdEmailAddress->Text      = Ur->EmailAddress;
   EdPassword->Text          = Ur->UserPassword;
   EdFirstName->Text         = Ur->FirstName;
   EdOtherInits->Text        = Ur->OtherInits;
   EdSurname->Text           = Ur->Surname;
   EdWebAddress->Text        = Ur->WebAddress;
   EdAffliation->Text        = Ur->Affiliation;
   EdDisplayName->Text       = Ur->DisplayName;

   AnsiString Query = "Select SmallDesc from Institutions where Institution_ID =";
   Query += IntToStr(Ur->InstitutionID);
   QryModUser->Close();
   QryModUser->SQL->Clear();
   QryModUser->SQL->Add(Query);
   try {
   QryModUser->Open();
     Ur->ItemIndex = CbxInstitution->Items->IndexOf (
            QryModUser->FieldByName("SmallDesc")->AsString);
```

```
     QryModUser->Close();
   } catch ( Exception& er) {
      ShowMessage(er.Message);
      LblError->Caption = EdEmailAddress->Text + " Database failed";
      BtnUpdate->Enabled = false;
      BtnPrior->Enabled=false;
      BtnNext->Enabled=false;
      DbNotFailed=false;
   }

   CbxInstitution->ItemIndex = Ur->ItemIndex;
   CbxUserRole->ItemIndex    = Ur->UserRole;

   EdStartDate->Text =     Ur->StartDate.DateString();
   EdRenewalDate->Text     = Ur->RenewalDate.DateString();
   BtnUpdate->Enabled = false;
//   EdEmailAddress->SetFocus();
   SetUpInProgress=false;

}
//---------------------------------------------------------------------------

void
TDlgModifyUser::SetUp(UserRecord* ur,const AnsiString& DbName){
   SetUpInProgress=true;
   DbNotFailed=true;
   Ur=ur;
   QryModUser->DatabaseName=DbName;

   EdPassword->EditMask =   EDMPassword;
   EdEmailAddress->MaxLength=LengthEmailaddress;
   EdPassword->MaxLength=LengthPassword;
   EdFirstName->MaxLength=LengthFirstName;
   EdOtherInits->MaxLength=LengthOtherInits;
   EdSurname->MaxLength=LengthSurname;
   EdWebAddress->MaxLength=LengthWebAddress;
   EdAffliation->MaxLength=LengthAffliation;
   EdDisplayName->MaxLength=LengthDisplayName;
   // load the institution list
   QryModUser->SQL->Clear();
   QryModUser->SQL->Add("Select SmallDesc from Institutions");
   try {
     QryModUser->Open();
     while (!QryModUser->Eof) {
       CbxInstitution->Items->Add(
         QryModUser->FieldByName("SmallDesc")->AsString);
       QryModUser->Next();
     }
   } catch ( Exception& er) {
      ShowMessage(er.Message);
      LblError->Caption = EdEmailAddress->Text + " SetUp failed";
      BtnUpdate->Enabled = false;
      BtnPrior->Enabled=false;
      BtnNext->Enabled=false;
      DbNotFailed=false;
   }

   QryModUser->Close();

}
//---------------------------------------------------------------------------
```

```
void __fastcall TDlgModifyUser::EdRenewalDateExit(TObject *Sender)
{
   TEdit* ted = dynamic_cast<TEdit*>(Sender);
   LblError->Caption ="";
//   if (ted->Text.Length() < 1 ) return;
   try {
      TDateTime test(ted->Text);
   } catch (Exception& E) {
     LblError->Caption ="The date is not valid";
     // display message
     ted->SetFocus();
     BtnUpdate->Enabled=false;
     return;
   }
   SetUpdateButton();
}
//---------------------------------------------------------------------------

void __fastcall TDlgModifyUser::CancelBtnClick(TObject *Sender)
{
   if (BtnUpdate->Enabled) {
      if (MessageDlg("Update Record?", mtConfirmation,
              TMsgDlgButtons() << mbYes << mbNo, 0) == mrYes) {
       BtnUpdateClick(Sender);
       if (BtnUpdate->Enabled) return; // failed
       }
   }
   ModalResult=mrCancel;

}
//---------------------------------------------------------------------------


void __fastcall TDlgModifyUser::SetUpdateButton(){

  BtnUpdate->Enabled = false;
  if (EdEmailAddress->Text.Trim().Length() < 1) return;

  if (( EdEmailAddress->Text != Ur->EmailAddress) ||
      (EdPassword->Text.Trim() != Ur->UserPassword)      ||
      (EdFirstName->Text != Ur->FirstName)        ||
      (EdOtherInits->Text != Ur->OtherInits)      ||
      (EdSurname->Text != Ur->Surname)            ||
      (EdWebAddress->Text != Ur->WebAddress)      ||
      (EdAffliation->Text != Ur->Affiliation)     ||
      (EdDisplayName->Text != Ur->DisplayName)    ||
      ( CbxInstitution->ItemIndex!=Ur->ItemIndex) ||
      (CbxUserRole->ItemIndex != Ur->UserRole)    ||
      (EdStartDate->Text != Ur->StartDate.DateString()) ||
      (EdRenewalDate->Text != Ur->RenewalDate.DateString()) )

          { BtnUpdate->Enabled=true; return; }
}

void __fastcall TDlgModifyUser::CbxUserRoleExit(TObject *Sender)
{
   SetUpdateButton();
}
//---------------------------------------------------------------------------
```

APPENDIX D – MODULE DLGUSERMODIFY

Module Name – DlgVwArticles

DlgVwArticles.h

```
//---------------------------------------------------------------------------
#ifndef DlgVwArticlesH
#define DlgVwArticlesH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBGrids.hpp>
#include <DBTables.hpp>
#include <Grids.hpp>
#include <DBCtrls.hpp>
#include <Menus.hpp>
//---------------------------------------------------------------------------
class TDlgFrmViewArticles : public TForm
{
__published:
        TTable *TblArticles;
        TDataSource *DsArticles;
        TDBGrid *DBGridArticles;
        TPanel *Panel1;
        TButton *BtnAdd;
        TButton *BtnKeyWords;
        TButton *BtnMathCodes;
        TButton *BtnModify;
        TButton *BtnAuthors;
        TButton *BtnDownLoads;
        TButton *CancelBtn;
        TDBNavigator *DBNavigator1;
        TQuery *QryTblArticles;
        TPopupMenu *PMnuFilter;
        TMenuItem *ChangeFilter1;
        TMenuItem *RemoveFilter1;
        void __fastcall BtnDownLoadsClick(TObject *Sender);
        void __fastcall BtnModifyClick(TObject *Sender);
        void __fastcall BtnAddClick(TObject *Sender);
        void __fastcall BtnAuthorsClick(TObject *Sender);
        void __fastcall BtnKeyWordsClick(TObject *Sender);
        void __fastcall BtnMathCodesClick(TObject *Sender);
        void __fastcall DBGridArticlesTitleClick(TColumn *Column);
        void __fastcall ChangeFilter1Click(TObject *Sender);
        void __fastcall DBGridArticlesColEnter(TObject *Sender);
        void __fastcall RemoveFilter1Click(TObject *Sender);
private:
        void __fastcall ChangeSortColumn();
        AnsiString ColumnFieldName;
        AnsiString FilterText;
        AnsiString FilterColumn;
        AnsiString SortColumn;
        static char* BaseQry;
public:
      virtual __fastcall TDlgFrmViewArticles(TComponent* AOwner);
      void SetUp(const AnsiString& dbname);
};
//---------------------------------------------------------------------------
extern PACKAGE TDlgFrmViewArticles *DlgFrmViewArticles;
//---------------------------------------------------------------------------
#endif
```

DlgVwArticles.Cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgVwArticles.h"
#include "DlgArticleDownloads.h"
#include "DlgModArticles.h"
#include "DlgAddAuthArt.h"
#include "DlgKeyWords.h"
//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TDlgFrmViewArticles *DlgFrmViewArticles;
//---------------------------------------------------------------------------
__fastcall TDlgFrmViewArticles::TDlgFrmViewArticles(TComponent* AOwner)
      : TForm(AOwner)
{
}
//---------------------------------------------------------------------------
void
TDlgFrmViewArticles::SetUp(const AnsiString& dbname) {
   QryTblArticles->DatabaseName = dbname;
   SortColumn="Article_Id";
   ColumnFieldName = "Article_Id";
   ChangeSortColumn();
}
void __fastcall TDlgFrmViewArticles::BtnDownLoadsClick(TObject *Sender)
{
// Work the down loads
   TDlgFrmDownloads * Dlg;
   try {
     Dlg = new TDlgFrmDownloads(this);
   } catch (Exception& e) { ShowMessage(e.Message); return; }
   Dlg->SetUp(QryTblArticles->FieldByName("Article_id")->AsString,
              QryTblArticles->DatabaseName);
   Dlg->ShowModal();
   delete Dlg;

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::BtnModifyClick(TObject *Sender)
{
   TDlgModifyArticle * Dlg;
   try {
      Dlg= new TDlgModifyArticle(this);
   } catch (Exception& e) { ShowMessage(e.Message); return; }
   Dlg->SetUp(DsArticles,QryTblArticles->DatabaseName,true);

   Dlg->ShowModal();
   delete Dlg;
   ChangeSortColumn();
```

```
}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::BtnAddClick(TObject *Sender)
{
   TDlgModifyArticle * Dlg;
   try {
      Dlg= new TDlgModifyArticle(this);
   } catch (Exception& e) { ShowMessage(e.Message); return; }
   Dlg->Caption = "Add an article";
   Dlg->SetUp(DsArticles,QryTblArticles->DatabaseName,false);
   Dlg->ShowModal();
   delete Dlg;
   ChangeSortColumn();
   QryTblArticles->Last();

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::BtnAuthorsClick(TObject *Sender)
{
// here we load up author details
   TFrmDlgAddAuthArt * Dlg;
   try {
      Dlg= new TFrmDlgAddAuthArt(this);
   } catch (Exception& e) { ShowMessage(e.Message); return; }
   Dlg->SetUp(QryTblArticles->SessionName,QryTblArticles->DatabaseName,
         QryTblArticles->Fields->FieldByName("Article_id")->AsString);
   Dlg->ShowModal();
   delete Dlg;
   DBGridArticles->Refresh();
}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::BtnKeyWordsClick(TObject *Sender)
{
// keywords
   TFrmChangeKeyWords * Dlg;
   try {
      Dlg= new TFrmChangeKeyWords(this);
   } catch (Exception& e) { ShowMessage(e.Message); return; }
   AnsiString caption = "Keywords for ";
   caption += QryTblArticles->Fields->FieldByName("ArticleTitle")->AsString;
   Dlg->Caption = caption;
   Dlg->SetUp(QryTblArticles->SessionName,QryTblArticles->DatabaseName,
         QryTblArticles->Fields->FieldByName("Article_id")->AsString,true);
   Dlg->ShowModal();
   delete Dlg;
   DBGridArticles->Refresh();

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::BtnMathCodesClick(TObject *Sender)
{
   TFrmChangeKeyWords * Dlg;
   try {
      Dlg= new TFrmChangeKeyWords(this);
   } catch (Exception& e) { ShowMessage(e.Message); return; }
   AnsiString caption = "Math Codes for ";
   caption += QryTblArticles->Fields->FieldByName("ArticleTitle")->AsString;
   Dlg->Caption = caption;
   Dlg->LblKeyWord->Caption = "New math code";
   Dlg->LblListBox->Caption = "Math Codes";
   Dlg->SetUp(QryTblArticles->SessionName,QryTblArticles->DatabaseName,
         QryTblArticles->Fields->FieldByName("Article_id")->AsString,false);
   Dlg->ShowModal();
   delete Dlg;
   DBGridArticles->Refresh();

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::DBGridArticlesTitleClick(
      TColumn *Column)
{
// title click signifies the sort column
   AnsiString col= Column->FieldName;
   AnsiString mes = "Change sort order to " + col + "?";
   if (MessageDlg(mes, mtConfirmation, TMsgDlgButtons()
         << mbYes << mbNo, 0) != mrYes) return;
   SortColumn = col;
   ChangeSortColumn();
}
//---------------------------------------------------------------------------
char* TDlgFrmViewArticles::BaseQry= "Select
article_Id,ArticleTitle,ReceiveDate,AcceptDate,Editor_Id,Published,Available from
articles ";
//---------------------------------------------------------------------------
void __fastcall TDlgFrmViewArticles::ChangeSortColumn(){
   try {
   QryTblArticles->Close();
   QryTblArticles->SQL->Clear();
   QryTblArticles->SQL->Add(BaseQry);
   if (FilterText.Length() > 0) {
     QryTblArticles->SQL->Add(" where " + FilterColumn + " like '%" + FilterText
+"%'");
   }
   QryTblArticles->SQL->Add(" Order by " + SortColumn );
   QryTblArticles->Open();
   } catch (Exception& e) {
     ShowMessage(e.Message);
     BtnAdd->Enabled =false;
     BtnModify->Enabled =false;
     BtnAuthors->Enabled =false;
     BtnKeyWords->Enabled =false;
     BtnMathCodes->Enabled =false;
     BtnDownLoads->Enabled =false;
   }

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::ChangeFilterlClick(TObject *Sender)
{
// change the filter text
   AnsiString col= ColumnFieldName;
   if (col != FilterColumn) {
     FilterText = "";
   }
   AnsiString InputString=FilterText;
   if (InputQuery("Set Filter Text for " + col, "Enter filter ", InputString)){
     if (InputString.Trim().Length() > 0) {
       FilterColumn=col;
       FilterText=InputString.Trim();
       ChangeSortColumn();
     }
```

```
    }

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::DBGridArticlesColEnter(
      TObject *Sender)
{
   ColumnFieldName = DBGridArticles->Columns->Items [ DBGridArticles->SelectedIndex]
                     ->FieldName;

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewArticles::RemoveFilter1Click(TObject *Sender)
{
// Clear the filter text
   FilterText = "";
   FilterColumn="";
   ChangeSortColumn();
}
//---------------------------------------------------------------------------
```

Appendix D – Module DlgVwArticles

Module Name – DlgVwUsers

DlgVwUsers.h

```
//---------------------------------------------------------------------------
#ifndef DlgVwUsersH
#define DlgVwUsersH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBGrids.hpp>
#include <DBTables.hpp>
#include <Grids.hpp>
#include <DBCtrls.hpp>
#include <Menus.hpp>
//---------------------------------------------------------------------------
class TDlgFrmViewUsers : public TForm
{
__published:
        TDBGrid *DBGrid1;
        TDataSource *DSJipam;
        TDBNavigator *DBNavigator1;
        TPanel *Panel1;
        TButton *BtnAddUser;
        TButton *BtnModify;
        TButton *BtnDelete;
        TButton *CancelBtn;
        TQuery *QryUsers;
        TPopupMenu *GridMenu;
        TMenuItem *ChangeFilter1;
        TMenuItem *RemoveFilter1;
        void __fastcall BtnAddUserClick(TObject *Sender);
        void __fastcall BtnModifyClick(TObject *Sender);
        void __fastcall BtnDeleteClick(TObject *Sender);
        void __fastcall DBGrid1TitleClick(TColumn *Column);
        void __fastcall ChangeFilter1Click(TObject *Sender);
        void __fastcall RemoveFilter1Click(TObject *Sender);
        void __fastcall DBGrid1ColEnter(TObject *Sender);
private:
        void __fastcall ChangeSortColumn();
        AnsiString ColumnFieldName;
        AnsiString FilterText;
        AnsiString FilterColumn;
        AnsiString SortColumn;
        static char* BaseQry;
public:
       virtual __fastcall TDlgFrmViewUsers(TComponent* AOwner);
        void SetUp(AnsiString& DbName);

};
//---------------------------------------------------------------------------
extern PACKAGE TDlgFrmViewUsers *DlgFrmViewUsers;
//---------------------------------------------------------------------------
#endif
```

DlgVwUsers.Cpp

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgVwUsers.h"
#include "DlgAddUser.h"
#include "DlgUserModify.h"

//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TDlgFrmViewUsers *DlgFrmViewUsers;
//---------------------------------------------------------------------------
__fastcall TDlgFrmViewUsers::TDlgFrmViewUsers(TComponent* AOwner)
      : TForm(AOwner)
{

}
//---------------------------------------------------------------------------
void
TDlgFrmViewUsers::SetUp(AnsiString& DbName){
//    DTblUsers->DatabaseName = DbName;
//    DTblUsers->Open();
    QryUsers->DatabaseName = DbName;
    SortColumn="Surname";
    ChangeSortColumn();
}


void __fastcall TDlgFrmViewUsers::BtnAddUserClick(TObject *Sender)
{
// add another user
   TDlgFmAddUser * Dlg = new TDlgFmAddUser(this);
   Dlg->SetUp(QryUsers->DatabaseName);
   Dlg->ShowModal();
   delete Dlg;
   QryUsers->Last();
   DBGrid1->Refresh();

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewUsers::BtnModifyClick(TObject *Sender)
{
//modify a user
    UserRecord CurrentUser;
    TDlgModifyUser * Dlg = new TDlgModifyUser(this);
    Dlg->SetUp(&CurrentUser,QryUsers->DatabaseName);
    while (true) {
      Dlg->BtnNext->Enabled=true;
      Dlg->BtnPrior->Enabled=true;
      if (QryUsers->Eof) {
//        DTblUsers->Prior();
        Dlg->BtnNext->Enabled=false;
      }
      if (QryUsers->Bof ) {
//        DTblUsers->Next();
        Dlg->BtnPrior->Enabled=false;
      }
```

```
      CurrentUser.User_id       = QryUsers->Fields->FieldByName("User_id")->AsInteger;
      CurrentUser.EmailAddress  = QryUsers->Fields->FieldByName("EmailAddress")-
>AsString.Trim();
      CurrentUser.UserPassword  = QryUsers->Fields->FieldByName("UserPassword")-
>AsString.Trim();
      CurrentUser.FirstName     = QryUsers->Fields->FieldByName("FirstName")-
>AsString.Trim();
      CurrentUser.OtherInits    = QryUsers->Fields->FieldByName("OtherInits")-
>AsString.Trim();
      CurrentUser.Surname       = QryUsers->Fields->FieldByName("Surname")-
>AsString.Trim();
      CurrentUser.WebAddress    = QryUsers->Fields->FieldByName("WebAddress")-
>AsString.Trim();
      CurrentUser.Affiliation   = QryUsers->Fields->FieldByName("Affiliation")-
>AsString.Trim();
      CurrentUser.DisplayName   = QryUsers->Fields->FieldByName("DisplayName")-
>AsString.Trim();
      CurrentUser.InstitutionID = QryUsers->Fields->FieldByName("Institution_ID")-
>AsInteger;
      CurrentUser.UserRole      = QryUsers->Fields->FieldByName("UserRole")-
>AsInteger;
      CurrentUser.StartDate     = QryUsers->Fields->FieldByName("StartDate")-
>AsDateTime;
      CurrentUser.RenewalDate   = QryUsers->Fields->FieldByName("RenewalDate")-
>AsDateTime;
      Dlg->SetUpRecord();
      if (Dlg->ShowModal()== mrCancel) break;
// Last accessed will not be modified
//      QryUsers->Refresh();
      if (Dlg->Direction==2) QryUsers->Next();
      if (Dlg->Direction==1) QryUsers->Prior();
    }
    delete Dlg;
//    QryUsers->Refresh();
}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewUsers::BtnDeleteClick(TObject *Sender)
{
//delete a user
   try {
     QryUsers->Delete();
     DBGrid1->Refresh();
   } catch (Exception& er) {
     ShowMessage(er.Message);

   }
}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewUsers::DBGrid1TitleClick(TColumn *Column)
{
// sort the column
   AnsiString col= Column->FieldName;
   AnsiString mes = "Change sort order to " + col + "?";
   if (MessageDlg(mes, mtConfirmation, TMsgDlgButtons()
         << mbYes << mbNo, 0) != mrYes) return;
   SortColumn = col;
   ChangeSortColumn();
}
//---------------------------------------------------------------------------
char* TDlgFrmViewUsers::BaseQry= "select
user_id,EmailAddress,UserPassword,FirstName,OtherInits,Surname,DisplayName,WebAddress,
UserRole,Institution_id,Affiliation,StartDate,RenewalDate,LastAccessed from Users";
//---------------------------------------------------------------------------
```

```
void __fastcall TDlgFrmViewUsers::ChangeSortColumn(){
   QryUsers->Close();
   QryUsers->SQL->Clear();
   QryUsers->SQL->Add(BaseQry);
   if (FilterText.Length() > 0) {
     QryUsers->SQL->Add(" where " + FilterColumn + " like '%" + FilterText +"%'");
   }
   QryUsers->SQL->Add(" Order by " + SortColumn );
   QryUsers->Open();

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewUsers::ChangeFilter1Click(TObject *Sender)
{
// change the filter text
   AnsiString col= ColumnFieldName;
   if (col != FilterColumn) {
     FilterText = "";
   }
   AnsiString InputString=FilterText;
   if (InputQuery("Set Filter Text for " + col, "Enter filter ", InputString)){
     if (InputString.Trim().Length() > 0) {
       FilterColumn=col;
       FilterText=InputString.Trim();
       ChangeSortColumn();
     }
   }

}
//---------------------------------------------------------------------------

void __fastcall TDlgFrmViewUsers::RemoveFilter1Click(TObject *Sender)
{
// Clear the filter text
   FilterText = "";
   FilterColumn="";
   ChangeSortColumn();
}
//---------------------------------------------------------------------------


void __fastcall TDlgFrmViewUsers::DBGrid1ColEnter(TObject *Sender)
{
   ColumnFieldName = DBGrid1->Columns->Items [ DBGrid1->SelectedIndex]
                       ->FieldName;
}
//---------------------------------------------------------------------------
```

Module Name - DlgVwVolumes

DlgVwVolumes.h

```
//---------------------------------------------------------------------------
#ifndef DlgVwVolumesH
#define DlgVwVolumesH
//---------------------------------------------------------------------------
#include <vcl\System.hpp>
#include <vcl\Windows.hpp>
#include <vcl\SysUtils.hpp>
#include <vcl\Classes.hpp>
#include <vcl\Graphics.hpp>
#include <vcl\StdCtrls.hpp>
#include <vcl\Forms.hpp>
#include <vcl\Controls.hpp>
#include <vcl\Buttons.hpp>
#include <vcl\ExtCtrls.hpp>
#include <Db.hpp>
#include <DBTables.hpp>
//---------------------------------------------------------------------------
class TFrmVwVolumes : public TForm
{
__published:
        TButton *CancelBtn;
        TBevel *Bevel1;
         TLabel *Label1;
         TListBox *LbxVolumes;
         TButton *BtnArticles;
         TButton *BtnNew;
         TQuery *QryGeneral;
         TButton *BtnChange;
         void __fastcall LbxVolumesClick(TObject *Sender);
         void __fastcall BtnArticlesClick(TObject *Sender);
         void __fastcall BtnNewClick(TObject *Sender);
         void __fastcall BtnChangeClick(TObject *Sender);
private:
   AnsiString DatabaseName;
   void __fastcall ClearObjects(TListBox*);
   void __fastcall SetUpListBox();

public:
        virtual __fastcall TFrmVwVolumes(TComponent* AOwner);
        virtual __fastcall ~TFrmVwVolumes();
         void __fastcall SetUp(const AnsiString dbname);

};
//---------------------------------------------------------------------------
extern PACKAGE TFrmVwVolumes *FrmVwVolumes;
//---------------------------------------------------------------------------
#endif
```

DlgVwVolumes.Cpp

```
//---------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DlgVwVolumes.h"
#include "DlgAddArtToVolume.h"
#include "DlgNewVolume.h"
//---------------------------------------------------------------------------
#pragma resource "*.dfm"
TFrmVwVolumes *FrmVwVolumes;
//---------------------------------------------------------------------------
__fastcall TFrmVwVolumes::TFrmVwVolumes(TComponent* AOwner)
        : TForm(AOwner)
{
}
//---------------------------------------------------------------------------
__fastcall TFrmVwVolumes::~TFrmVwVolumes()
{
   ClearObjects(LbxVolumes);
}
//---------------------------------------------------------------------------
void __fastcall TFrmVwVolumes::ClearObjects(TListBox* tlb){
  for (int i=0; i<tlb->Items->Count; i++){
     TMVolObject* tob = (TMVolObject*) tlb->Items->Objects[i];
     delete tob;
  }
}
//---------------------------------------------------------------------------
void __fastcall TFrmVwVolumes::SetUp(const AnsiString dbname){
  DatabaseName=dbname;
  QryGeneral->DatabaseName=dbname;
  SetUpListBox();

}
//---------------------------------------------------------------------


void __fastcall TFrmVwVolumes::LbxVolumesClick(TObject *Sender)
{
  BtnArticles->Enabled =true;
  BtnChange->Enabled=true;
}
//---------------------------------------------------------------------------

void __fastcall TFrmVwVolumes::BtnArticlesClick(TObject *Sender)
{
// view articles on volume
   TMVolObject* tvb = (TMVolObject*) LbxVolumes->Items->Objects[
                          LbxVolumes->ItemIndex];
   TFrmAddArtToVolume * Dlg = new TFrmAddArtToVolume(this);

   try {
   Dlg->Caption = LbxVolumes->Items->Strings[LbxVolumes->ItemIndex] + " Articles";
   Dlg->SetUp(DatabaseName,tvb->Volume_id);
   Dlg->ShowModal();
   } catch (Exception& e) {
     ShowMessage(e.Message);
   }
   delete Dlg;


}
//---------------------------------------------------------------------------

void __fastcall TFrmVwVolumes::BtnNewClick(TObject *Sender)
{
```

```
// add another volume
    TFrmNewVolume * Dlg = new TFrmNewVolume(this);
    TMVolObject* tvb = new TMVolObject;
    if (LbxVolumes->Items->Count > 0) {
      *tvb = *((TMVolObject*) LbxVolumes->Items->Objects[
                          LbxVolumes->Items->Count - 1 ]);
//      tvb->ISSUE_NO++;
    } else {
       tvb->DateOfIssue= Date();
    }
   try {
     Dlg->SetUp(DatabaseName,tvb,false);
     if (Dlg->ShowModal() == mrOk ){
        SetUpListBox();
     }
   } catch (Exception& e) {
     ShowMessage(e.Message);
   }
   delete tvb;
   delete Dlg;
}
//---------------------------------------------------------------------------

void __fastcall TFrmVwVolumes::BtnChangeClick(TObject *Sender)
{
// change volume
    if (LbxVolumes->ItemIndex  < 0) return;

    TFrmNewVolume * Dlg = new TFrmNewVolume(this);
    AnsiString caption = "Modify ";
    caption += LbxVolumes->Items->Strings [
                          LbxVolumes->ItemIndex ];
    Dlg->Caption = caption;
    TMVolObject* tvb;
    tvb = (TMVolObject*) LbxVolumes->Items->Objects[
                          LbxVolumes->ItemIndex ];
    try {
      Dlg->SetUp(DatabaseName,tvb,true);
      if (Dlg->ShowModal() == mrOk ){
         SetUpListBox();
      }
    } catch (Exception& e) {
      ShowMessage(e.Message);
    }
    delete Dlg;

}
//---------------------------------------------------------------------------
void __fastcall TFrmVwVolumes::SetUpListBox(){
   ClearObjects(LbxVolumes);
   LbxVolumes->Items->Clear();
  // fill the list box
  QryGeneral->Close();
  QryGeneral->SQL->Clear();
  QryGeneral->SQL->Add("Select Volume_id,VOLUME_NO,ISSUE_NO,ISSUEYear,ISSUEDate from
VOLUMES");
  try {
    QryGeneral->Open();
    while(!QryGeneral->Eof ) {
      TMVolObject* tvb = new TMVolObject;
        tvb -> Volume_id= QryGeneral->FieldByName("Volume_id")->AsInteger;
        tvb->VOLUME_NO =  QryGeneral->FieldByName("VOLUME_NO")->AsInteger;
        tvb->ISSUE_NO =   QryGeneral->FieldByName("ISSUE_NO")->AsInteger;
        tvb->ISSUEYear =  QryGeneral->FieldByName("ISSUEYear")->AsInteger;
        tvb->DateOfIssue = QryGeneral->FieldByName("ISSUEDate")->AsDateTime;
      AnsiString Vol = "Vol. " + IntToStr(tvb->VOLUME_NO);
      Vol += ", No. " + IntToStr(tvb->ISSUE_NO);
      Vol += ", " + IntToStr(tvb->ISSUEYear);
      LbxVolumes->Items->AddObject(Vol,tvb);
      QryGeneral->Next();
    }
  } catch (Exception& e) {
    ShowMessage(e.Message);
    BtnArticles->Enabled =false;
    BtnNew->Enabled =false;
    BtnChange->Enabled=false;
  }
  QryGeneral->Close();
}
//---------------------------------------------------------------------------
```

Module Name – JipamCommon

JipamCommon.h

```
#ifndef JIPAMCOMMON_H
#define JIPAMCOMMON_H

struct UserRecord {
  int User_id;
  AnsiString EmailAddress;
  AnsiString UserPassword;
  AnsiString FirstName;
  AnsiString OtherInits;
  AnsiString Surname;
  AnsiString WebAddress;
  AnsiString Affiliation;
  AnsiString DisplayName;
  int InstitutionID;
  int ItemIndex;
  int UserRole;
  TDateTime StartDate;
  TDateTime RenewalDate;
  TDateTime LastAccessed;
} ;
extern char* EDMPassword;
extern const int LengthEmailaddress;
extern const int LengthPassword;
extern const int LengthFirstName;
extern const int LengthOtherInits;
extern const int LengthSurname;
extern const int LengthDisplayName;
extern const int LengthWebAddress;
extern const int LengthAffliation;
extern const int LengthKeyword;
extern const int LengthMathCode;

extern const int LengthSmallDesc;
extern const int LengthAddressLine1;
extern const int LengthAddressLine2;
extern const int LengthAddressLine3;
extern const int LengthAddressLine4;
extern const int LengthAddressLine5;
extern const int LengthTelephone;
extern const int LengthFascimile;

class ArticleObj : public TObject {
private:
   AnsiString Article_Id;
   bool Moved;
public:
   ArticleObj() {Moved=false;}
   ArticleObj(AnsiString& aid) {Moved=false;Article_Id = aid;}
   ArticleObj(ArticleObj* t) {Article_Id = t->Article_Id; Moved=false; }
   AnsiString& GetArticleId() { return Article_Id; }
   bool GetMoved() { return Moved; }
   void SetArticleId(AnsiString aid) { Article_Id=aid; }
   void SetMoved() { Moved=true; }
   ArticleObj& operator=(ArticleObj& t) { Article_Id=t.Article_Id; return *this;}
};
class PrintTypeObj: public TObject {
private:
   int Printtype;
public:
  PrintTypeObj() {Printtype=-1; }
  PrintTypeObj(int pt) { Printtype = pt; }
  PrintTypeObj(PrintTypeObj* pto) {Printtype=pto->Printtype; }
  int GetPrintType() { return Printtype; }
};

class TMObject : public TObject {
private:
   int User_id;
   int Institution_id;
   int UserRole;
   int Year;
public:
   TMObject(int uid,int iid,int ur,int Yr=0){ User_id=uid;Institution_id=iid;
       UserRole=ur; Year =Yr; }
   TMObject(TMObject* t) { User_id=t->User_id ;Institution_id=t->Institution_id;
       UserRole=t->UserRole; Year = t->Year;  }
   TMObject() {User_id=0;Institution_id=0;UserRole=0;Year=0; }
   int GetUser_id() { return User_id; }
   int GetInstitution_id() { return Institution_id; }
   int GetUserRole() {return UserRole; }
   int GetYear() { return Year; }
   void SetInstitution(int iid) {Institution_id=iid;}
   TMObject& operator=(TMObject& t) { User_id=t.User_id
;Institution_id=t.Institution_id;
       UserRole=t.UserRole; Year = t.Year; return t;}
};
class TMVolObject : public TObject {
public:
  int Volume_id;
  int VOLUME_NO;
  int ISSUE_NO;
  int ISSUEYear;
  TDateTime DateOfIssue;
  TMVolObject() {Volume_id=1;VOLUME_NO=1;ISSUE_NO=1;ISSUEYear=2000;}

};
class TMArticleImageObj : public TObject {
public:
  short status; //1 = existing,2=new,3 =deleted
  int ImageNumber ;
  AnsiString SourceFileName;
  AnsiString SourceFileType;
  AnsiString AltText;
  bool AltTextChange;
  TMArticleImageObj() { status = 1; ImageNumber=0; AltTextChange=false; }

};
class TMEditorObj : public TObject {
public:
  int user_id;
  TMEditorObj() {user_id=0; }
};
class TMFullArticleObj : public TObject {
public:
  AnsiString Article_id;
  AnsiString Title;
  AnsiString ReceiveDate;
  AnsiString AcceptDate;
  int EditorId;
  bool Available;
  bool Published;
```

```
  AnsiString Abstract;
  TMFullArticleObj() { EditorId=0; Available=false; Published=false; }

};
#endif
```

Appendix D – Module JipamCommon

Module Name - JipamMain

JipamMain.h

```
//---------------------------------------------------------------------------

#ifndef JipammainH
#define JipammainH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//---------------------------------------------------------------------------
class TFrmMainForm : public TForm
{
__published:  // IDE-managed Components
        TButton *BtnViewUsers;
        TButton *BtnVwArticels;
        TButton *BtnVwVolumes;
        TButton *BtnLeave;
        TButton *BtnInstitutions;
        void __fastcall BtnViewUsersClick(TObject *Sender);
        void __fastcall BtnVwArticelsClick(TObject *Sender);
        void __fastcall BtnVwVolumesClick(TObject *Sender);
        void __fastcall BtnLeaveClick(TObject *Sender);
        void __fastcall BtnInstitutionsClick(TObject *Sender);
        void __fastcall FormShow(TObject *Sender);
private:      // User declarations
   void __fastcall SetUpDatabase(AnsiString DbName);
public:              // User declarations
        __fastcall TFrmMainForm(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE TFrmMainForm *FrmMainForm;
//---------------------------------------------------------------------------
#endif
```

JipamMain.Cpp

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "JipamCommon.h"
#include "DMJipam.h"
#include "Jipammain.h"
#include "DlgVwUsers.h"
#include "DlgVwArticles.h"
#include "DlgVwVolumes.h"
#include "DlgInstitutions.h"
AnsiString PrmDefaultDatabase = "JIPAM";
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TFrmMainForm *FrmMainForm;
//---------------------------------------------------------------------------
__fastcall TFrmMainForm::TFrmMainForm(TComponent* Owner)
        : TForm(Owner)
{
}
```

```
void __fastcall TFrmMainForm::SetUpDatabase(AnsiString DbName){

  TDatabase * tdb = DataModuleJipam->DbJipam;
  if (tdb->Connected) return;
  tdb->DatabaseName = DbName;
  TSession* ts = DataModuleJipam->SessionJipam;
  if (!ts->Active) ts->Open();
  int LoginTryCnt=3;
  while(LoginTryCnt> 0) {
    try {
    tdb->Open();
    } catch (Exception& e) {
      ShowMessage(e.Message);
    }
    if (tdb->Connected) return;
    LoginTryCnt--;
  }
  Close();
}
//---------------------------------------------------------------------------
void __fastcall TFrmMainForm::BtnViewUsersClick(TObject *Sender)
{
   TDlgFrmViewUsers * Dlg;
   try {
     Dlg = new TDlgFrmViewUsers(this);
   } catch (Exception & e) {
     ShowMessage(e.Message);
     return;
   }
   this->Hide();
   try {
   Dlg->SetUp(PrmDefaultDatabase);
   Dlg->ShowModal();
   } catch (Exception& e) {
     ShowMessage(e.Message);
   }
   this->Show();
   delete Dlg;

}
//---------------------------------------------------------------------------

void __fastcall TFrmMainForm::BtnVwArticelsClick(TObject *Sender)
{
   TDlgFrmViewArticles * Dlg;
   try {
   Dlg = new TDlgFrmViewArticles(this);
   } catch (Exception & e) {
     ShowMessage(e.Message);
     return;
   }
   this->Hide();
   try {
   Dlg->SetUp(PrmDefaultDatabase);
   Dlg->ShowModal();
   } catch (Exception& e) {
     ShowMessage(e.Message);
   }
   this->Show();
   delete Dlg;

}
```

```
void __fastcall TFrmMainForm::BtnVwVolumesClick(TObject *Sender)
{
   TFrmVwVolumes * Dlg;
   try {
   Dlg = new TFrmVwVolumes(this);
   } catch (Exception & e) {
     ShowMessage(e.Message);
     return;
   }
   this->Hide();
   try {
   Dlg->SetUp(PrmDefaultDatabase);
   Dlg->ShowModal();
   } catch (Exception& e) {
     ShowMessage(e.Message);
   }
   this->Show();
   delete Dlg;

}
//---------------------------------------------------------------------------

void __fastcall TFrmMainForm::BtnLeaveClick(TObject *Sender)
{
  Close();
}
//---------------------------------------------------------------------------

void __fastcall TFrmMainForm::BtnInstitutionsClick(TObject *Sender)
{
   TFrmInstitutions * Dlg;
   try {
   SetUpDatabase(PrmDefaultDatabase);
   Dlg = new TFrmInstitutions(this);
   } catch (Exception & e) {
     ShowMessage(e.Message);
     return;
   }
   this->Hide();
   try {
   Dlg->SetUp(PrmDefaultDatabase);
   Dlg->ShowModal();
   } catch (Exception& e) {
     ShowMessage(e.Message);
   }
   this->Show();
   delete Dlg;


}
//---------------------------------------------------------------------------

void __fastcall TFrmMainForm::FormShow(TObject *Sender)
{
   BtnViewUsers->Enabled=false;
   BtnVwArticels->Enabled=false;
   BtnVwVolumes->Enabled=false;
   BtnInstitutions->Enabled=false;
   SetUpDatabase(PrmDefaultDatabase);
   BtnViewUsers->Enabled=true;
   BtnVwArticels->Enabled=true;
   BtnVwVolumes->Enabled=true;
   BtnInstitutions->Enabled=true;
}
//---------------------------------------------------------------------------
```

Appendix D – Module JipamMain

Appendix E

JIPAM WEB SITE
Dynamic Database System

Administration Program Guide

ISSN 1443-5756

Journal of Inequalities in Pure and Applied Mathematics

Administration Program Guide

JOURNAL OF INEQUALITIES IN PURE AND APPLIED MATHEMATICS

Administration Program Guide

Trevor McGuckin
Phone 03 9743 5768

January 2001

School of Communications and Infomatics
Faculty of Engineering and Science

Table of Contents

Chapter 1

Introduction

ICON KEY

- JIPAM Install CD
- Series of Steps
- Valuable information

THis guide contains detailed information as a reference source for the JIPAM Administrator to initially setup the JIPAM Web System and then to cover its ongoing maintenance. Several features have been included into this written guide to assist in administering the JIPAM web site effectively. For instance, icons are shown in the left margin where appropriate to help navigate this guide.

JIPAM Web System

The JIPAM Web System (the JIPAM System) comprises the following three elements.

- A database component
- A web browser component
- An administration component

The JIPAM System is based on Microsoft SQL Server Version 7 database software and uses Microsoft Internet Information Server (IIS) as the web server. IIS uses Open Database Connectivity (ODBC) to retrieve information from the database. A JIPAM Administration Program (the administration component) has been developed to maintain the JIPAM System (See Chapter 2 onwards).

A JIPAM Installation CD-ROM has been provided (included as Appendix A) and full setup instructions for these three components follows.

Note

It is necessary for the Admin. Program to also use ODBC to connect to the database.

Creating the JIPAM Database

The JIPAM Database (called the database from now on) has been developed using Microsoft SQL Server Version 7 database software. It is the basis of the JIPAM System and is created using three SQL procedures followed by an NT Batch Process provided on the JIPAM Installation CD. The first two SQL procedures set up the database tables, the final SQL procedure loads real data into these tables while the NT batch file concludes loading the data by inserting the relevant PDF files and any associated image files.

Note

The email subsystem and the SQL server agent subsystem of the database software need to be enabled by the SQL Server Database Administrator.

The first SQL procedure assumes the presence of the JIPAM database. The database needs to be created by the SQL Server Database Administrator on the computer hosting the SQL database program.

1. Once the database exists, run the **Query Analyzer Program** provided with Microsoft SQL Server Version 7 and then login to the JIPAM database as administrator. Run the first SQL procedure file, **JIPAM/SQL/SETUP.SQL** from the JIPAM Installation CD using the Query Analyzer Program.

2. Run the second SQL procedure file, **JIPAM/SQL/PROCEDURES.SQL** from the JIPAM Installation CD using the Query Analyzer Program.

3. Run the third SQL file, **JIPAM/SQL/DATALOAD/DATALOAD.SQL** from the JIPAM Installation CD using the Query Analyzer Program to load the database with real data.

4. To finish the JIPAM database process, run the NT Batch file, **JIPAM/SQL/DATALOAD/LOADPDF.CMD** from the JIPAM Installation CD to load the database with the PDF files and any associated image files.

MS Internet Information Server

The MS Internet Information Server (IIS) administrator will need to create an application called **JIPAM.** The contents of the **JIPAM/JIPAM** directory are to be copied from the JIPAM Installation CD into the IIS **JIPAM** application directory. The copy process must also include all of subdirectories of **JIPAM/JIPAM** on the JIPAM Installation CD.

The IIS administrator will also need to create an ODBC system DSN called **JIPAM** (see the following section).

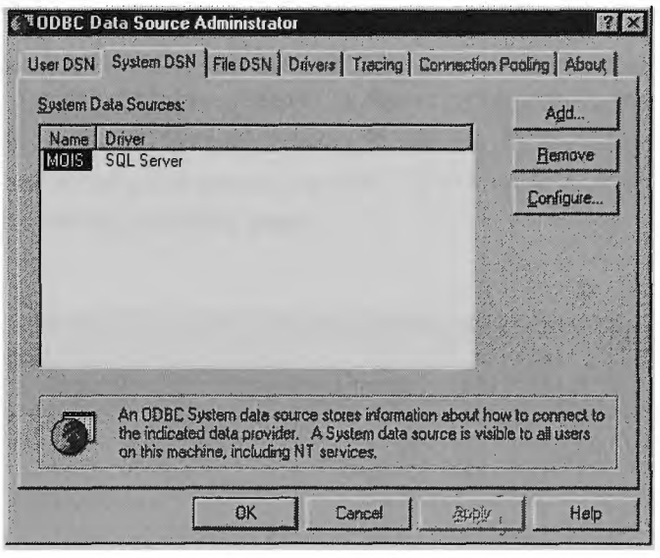
ODBC (Open Database Connectivity) Installation

The JIPAM administrator program (referred to as the admin program from now on) requires a data source based on the ODBC (Open Database Connectivity) driver (provided by Microsoft) to access the SQL Server 7 database. If the driver is not installed on the administrator's computer, refer to the SQL Server 7 documentation for details about installing the driver.

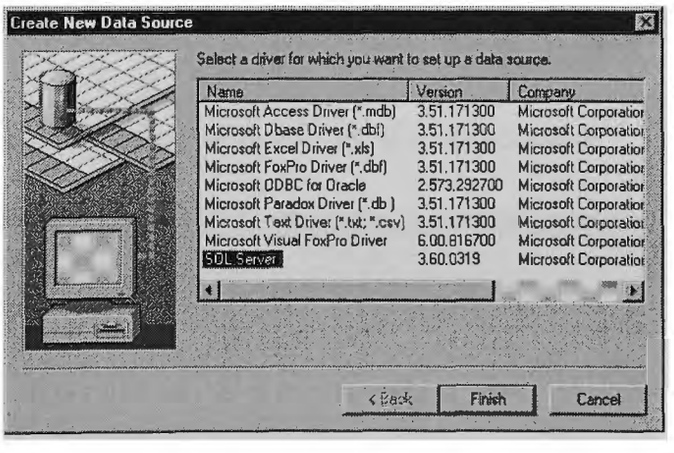
➢ To create a data source named JIPAM

1. In Control Panel, click the **ODBC Data Sources** icon.

2. Click **System DSN** tab. Select **SQL Server** and then click **Add.**

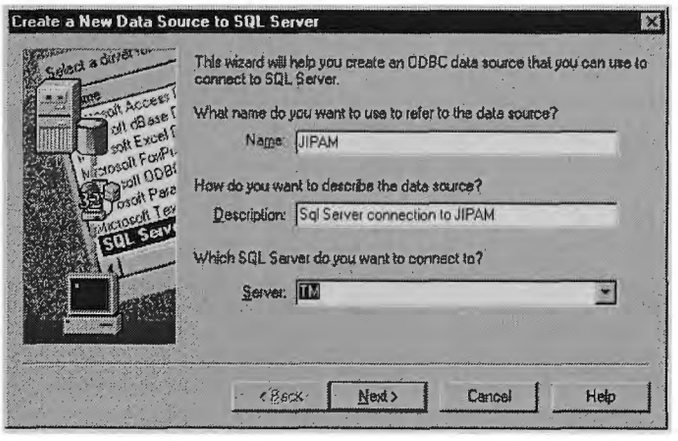


3. Select **SQL Server** from the Name list and click **Finish**.

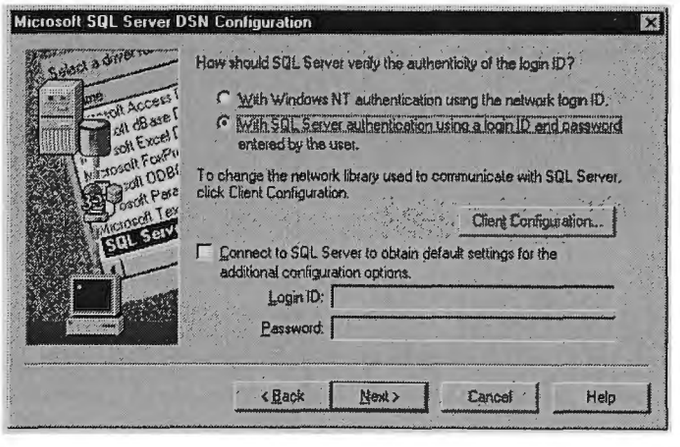


4. A wizard to create an ODBC data source is provided. Enter the data source name as **JIPAM** and the description as **SQL Server Connection to JIPAM**. Select the network address of the computer on which the JIPAM

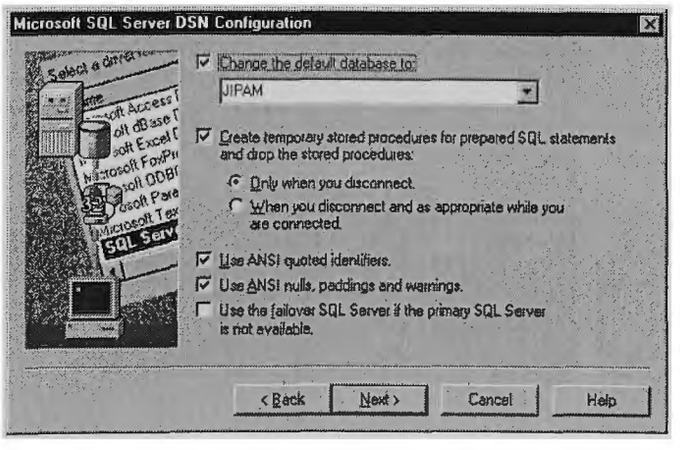
database will reside (the appropriate SQL server) from the drop down list. Click **Next**.



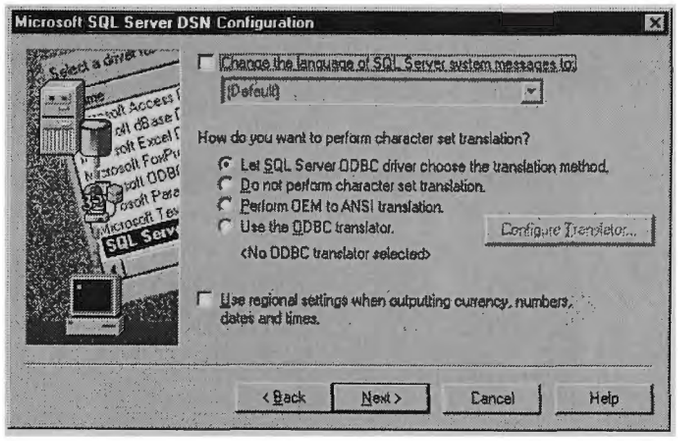
5. The next dialogue box should be completed after consultation with the SQL Server Database Administrator. It provides two options regarding the logon process to the JIPAM database. Given that the database is also accessible via the web, the second option will probably be sufficient. Complete the screen and click **Next**.



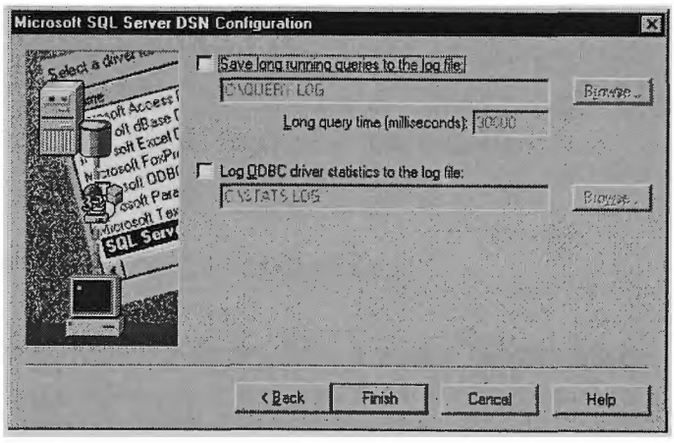
6. Complete the dialogue box as shown below and click **Next**.



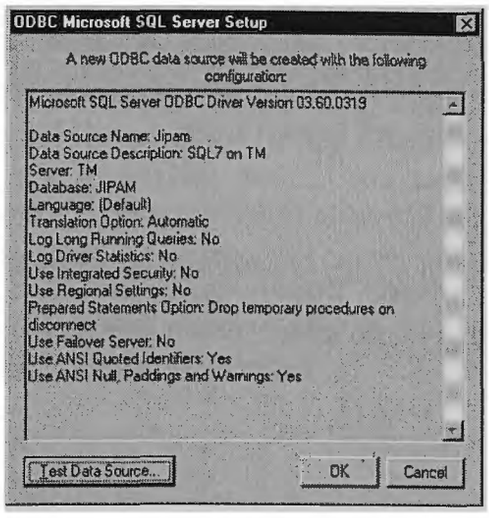
7. No changes are necessary, so again just click **Next**.



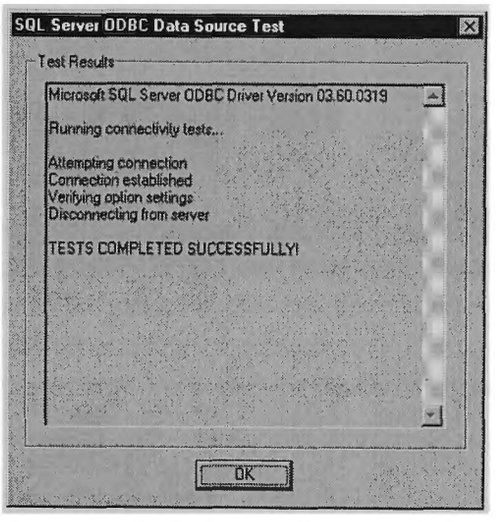
8. No changes are necessary in the final step to create a data source named JIPAM so click **Finish** in the dialogue box below.



9. On completion of the wizard, the specifications of the new ODBC data source are displayed. The specifications shown below are a test setup only and your configuration will be different.



10. Click **Test Data Source** to ascertain that the database can be accessed. Your details will be different to that shown below which is a test only. If your test is successful, click **OK** and the data source will be added to the System DSN table in the ODBC Manager. If unsuccessful, consult the SQL Database Administrator.



11. The creation of the ODBC interface is now complete.

Admin. Program Installation

The JIPAM Admin. Program allows the database to be maintained on an ongoing basis. Maintenance is undertaken based on the following four logical views of the database.

- Subscribers (online users (including authors and editors) who have registered to access the full text of articles published in JIPAM)
- Articles (articles published in all JIPAM volumes)
- Volumes (each issue of JIPAM is known as a volume)
- Institutions (subscribers provide institutional details of their institution.)

Run the file **JIPAM/INSTALL/SETUP.EXE** from the JIPAM Installation CD following the screen prompts to install the JIPAM Administration Program (called Admin. Program from now on) into the directory you require. A menu item called JIPAM Admin is added to the Programs menu as part of the Admin. Program installation process. It is accessed from the Start button on your desktop.

Ongoing maintenance of the JIPAM Web System can be undertaken after the Admin. Program has been successfully installed.

Chapter 2

The Admin. Program

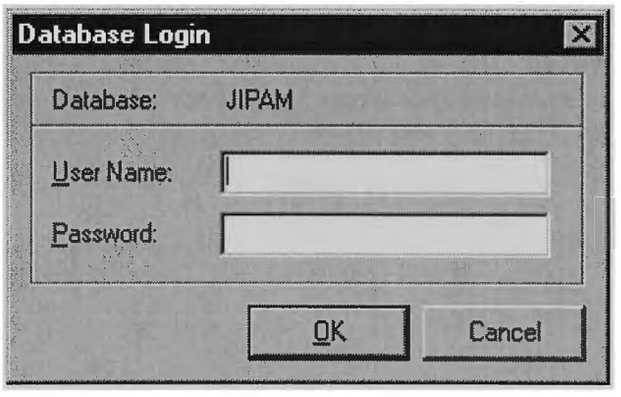
The Admin. Program is a management tool for the System Administrator to maintain the data in the JIPAM database. It is structured on four logical views of the data (for instance, the data associated with the subscribers, the articles, the volumes and the associated institutions) and options are provided for the administrator to view and/or modify the data.

Starting the Admin. Program

➢ **To login to the Admin. Program**

1. Click **Start** on your desktop. Click on **Programs** then click on the submenu item called **JIPAM Admin.**

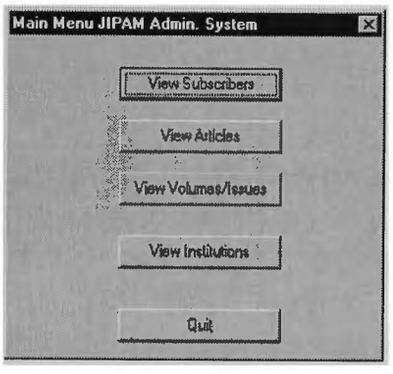
2. Login to the Admin. Program as the System Administrator by entering your **User Name** and **Password**.



Note

To undertake System Administrator functions, the login user name must have read, write, modify and execute permissions for the JIPAM database. When the database was originally created, a login identify with user name JPADMN and password JPADMN was created for system administration functions.

3. The **Main Screen JIPAM Admin. Program** displays the options available to the System Administrator in terms of maintaining the database.



4. The first four buttons allow specific sections of the database to be viewed and/or updated while **Quit** is used to exit the JIPAM Admin. Program.

Note

Detailed documentation relating to maintenance activities associated with each of the four views of the database are provided in the following four chapters.

Chapter 3

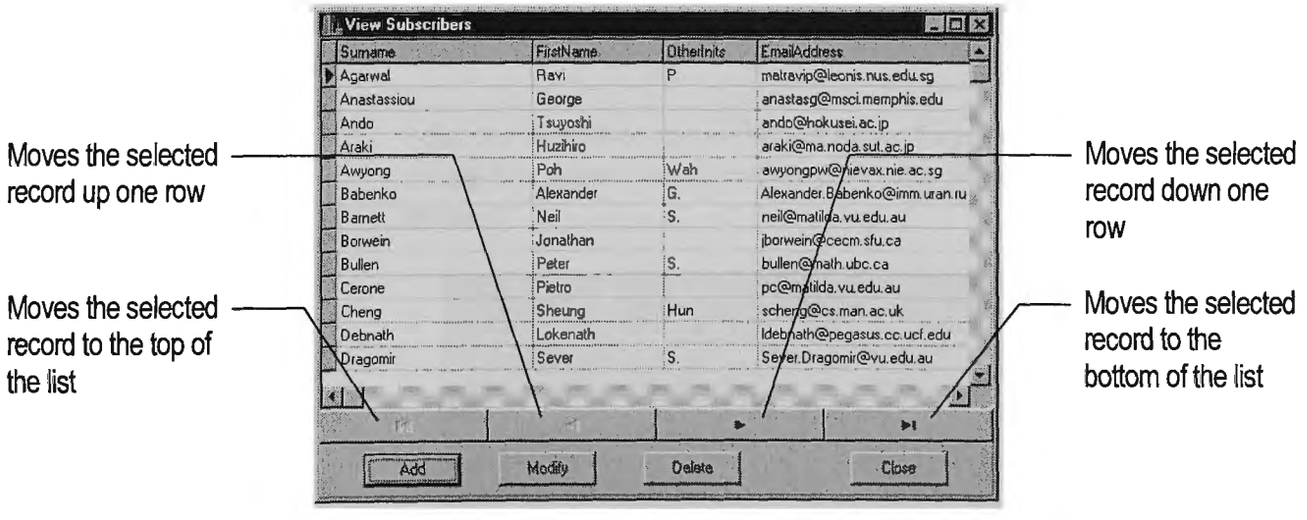
Subscribers

To gain access to the full text of articles published in JIPAM, it is necessary for users to be registered online as subscribers. This is a simple matter of providing relevant data such as an email address and login details that are stored within the JIPAM database. At this stage, subscriptions are free although that may change at some date in the future. The Admin. Program allows the JIPAM System Administrator access to subscriber details for maintenance purposes.

Subscriber Data Maintenance

In maintaining the subscriber-related data in the database, it is necessary to view the contents of the database.

Click **View Subscribers** on **Main Screen JIPAM Admin. System**. The **View Subscribers** screen provides easy access to the subscriber data.



The dialogue box is resizable to allow more or less detail to be shown in the table. Four navigation buttons are provided to allow the position of the selected subscriber's record data to be changed in the database table (see above).

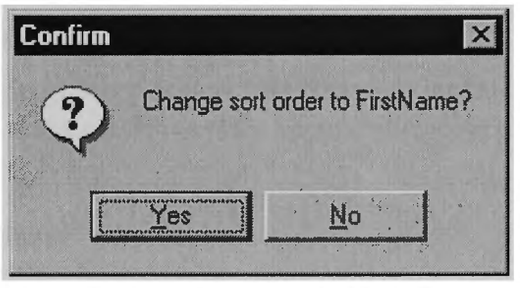
Maintenance of subscriber data is structured around the following three actions.

- Add a subscriber
- Modify a subscriber's data
- Delete a subscriber

Subscriber Search

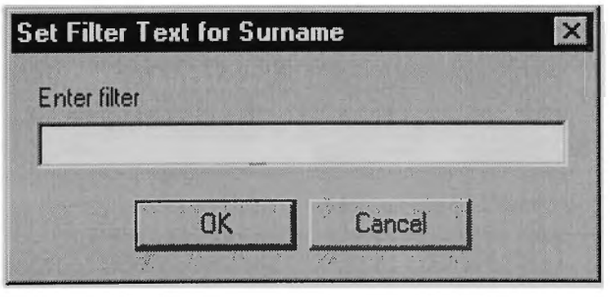
Two support facilities have been incorporated into the **View Subscribers** screen to assist in locating a specific subscriber's record within the database. They can be used singly or in combination.

Sort the subscriber table on any of the fields by clicking on the required field in the **Title Bar**. A confirmation message is displayed which identifies the sort criteria. Click **Yes** to change the order of the table to your selection or **No** to leave the table unchanged. The subscribers table will initially open in **Surname** order.



Search the subscriber table using filter text by right clicking on any column in the table although complex filters on more than one column are not currently supported. A pop up a menu appears with two options (1) Set filter and (2) Remove filter.

1. Selecting **Set filter** opens the **Set Filter Text for Surname** dialog box shown below which allows the filter text to be entered.



2. Click **OK** to apply the filter. The **View Subscribers** screen will then be displayed showing only the records that meet the filter criteria.

3. To remove the filter, right click again and select **Remove Filter.** (The **Cancel** button on the **Set Filter Text for**

Surname screen does not remove the filter.) Alternatively, close the **View Subscribers** screen and reopen it to return to the default setting with all subscribers records being shown in surname order (i.e. with no filters applied).

Note

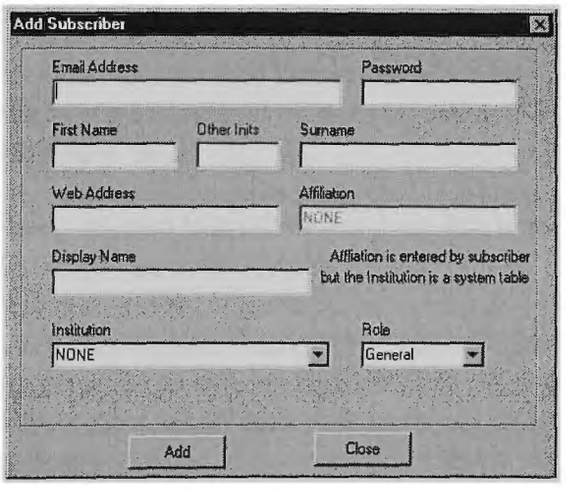
To list all subscribers who have a surname containing the text 'jones', right click on the surname column in the **View Subscribers** screen and enter the text 'jones' in the **Set Filter Text for Surname** entry box. The **View Subscribers** screen will list only those subscriber records that contain the text 'jones' anywhere in the surname field.

Add a Subscriber

Users can be manually registered as new subscribers in instances where the online registration process is unsuccessful or they are unable to use the online process.

➢ To add a subscriber

1. Click **View Subscribers** on **Main Screen JIPAM Admin. System**.
2. Click **Add** on the **View Subscribers** screen.
3. The **Add Subscriber** dialog box is displayed.



4. Fill in the required **data fields** (detailed explanations of the data fields are provided below).

    - *Email Address* is the email address of the user who is to be registered as a subscriber. This attribute identifies who the subscriber is to be and therefore must be unique within the database.

- *Password* is the subscriber's password (minimum length is four characters).

- *First Name, Other Inits* and *Surname* fields are self-explanatory. They can be left blank.

- *Web Address* is provided for those authors/editors who have their own web site and who would like users to know the URL. The field may be left blank.

- *Display Name* is the name that subscribers wish to have displayed on any screens produced in the web environment. The field is only applicable to authors and editors.

- *Affiliation* is the name that subscribers enter to identify their institution. Its primary purpose is to provide information to the System Administrator so that they can allocate the correct institution to authors and editors. The field is potentially important to authors and editors.

- *Institution* can only be selected from the displayed list. When users originally register as subscribers, the default entry in their **Institution** field is **None**. At the moment, the **Institution** field is only maintained (that is, changed from **None)** for authors and editors, not for all subscribers. Thus, the administrator needs to manually change the default entry (**None**) for all authors and editors based on the affiliation data they originally entered with their subscription information. If the institution required is not in the list, see Chapter 6 for details on adding a new institution into the database. On the **Add Subscribers** screen, the 'Affiliation' edit box is locked and changes to reflect the contents of the institution field.

- *Role* is selected from the **Role** drop down combination box. There are currently defined roles of *General, Author, Editor, Both* and *Admin*. All subscribers are assigned the *General* tag when they register online. The *Author, Editor* and *Both* roles are expected to be used more heavily when authors and editors can control the submission and acceptance of articles.

5. Click **Add** when all necessary data has been entered and this adds the new subscriber's details into the database. The addition will be successful provided the email address is unique. An error message will be displayed if the email address (the unique identifier) already exists.

6. Click **Close** to leave the subscribers table without adding any new subscribers to the database.

Modify a Subscriber

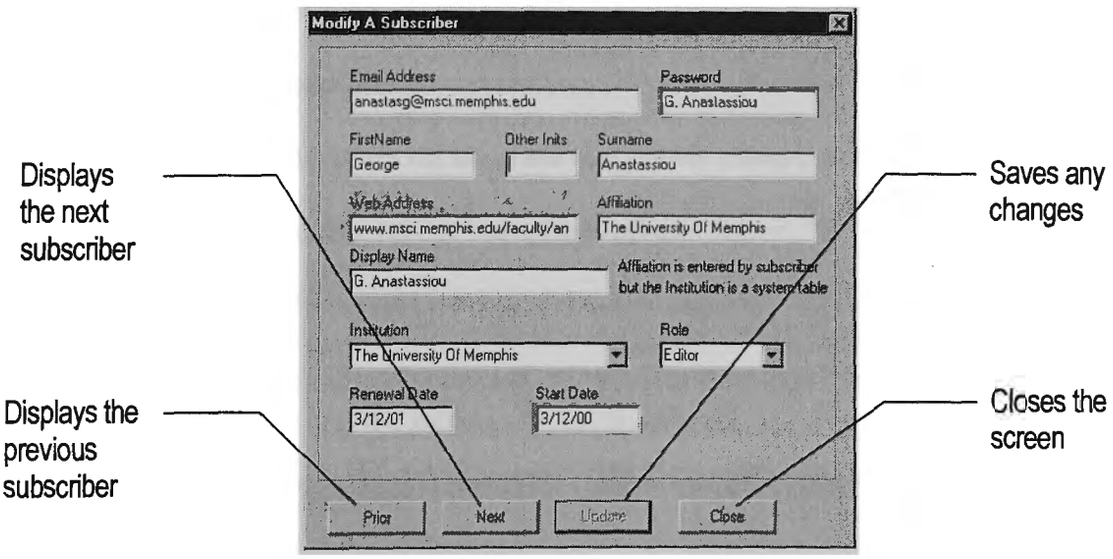
Although subscribers are able to modify most of their own details online, two data fields can only be changed by the System Administrator. These are the subscriber's role and the institution to which they are currently assigned. Rights to change this data have been restricted to the administrator to provide higher levels of security for the JIPAM Web System.

- **Role.** This field is used in a number of SQL procedures that determine who are JIPAM authors and editors. It is also used to enable the Display Name field for authors and editors when they want to change their login details from the web. The Admin role enables a subscriber (who is also the administrator) to access the Web Reports screen. Access to the Web Reports screen and other administration functions is restricted for security purposes.
- **Institution.** This field is only accessible to the administrator. Refer to Chapter 6 for more information

In addition, if a subscriber cannot remember their password (and they do not use the automatic email-back facility provided in the web login screen), the System Administrator has access to their password on the **Modify Subscriber** screen.

➢ To modify a subscriber

1. Click **View Subscribers** on **Main Screen JIPAM Admin. System**. Select the required subscriber from the list.
2. Click **Modify** on the **View Subscribers** screen.
3. The **Modify Subscriber** dialog box is displayed complete with the selected subscriber's data.



4. Fill in the required **data fields** (detailed explanations of the data fields are provided below).

Note

All comments related to the contents of the fields described in the previous section on adding new subscribers also apply to modifying subscriber data.

- **Renewal Date** is twelve months from the subscription date. If the field needs to be changed, the date must be a valid date otherwise an error message will be displayed and the cursor will remain in the edit box.

- **Start Date** is the date when the subscriber initially subscribed to the JIPAM Web System.

5. Click **Update** to save the modified data. At this point, the subscriber's record in the database is changed. However, due to system and program limitations, the **Modify Subscribers** screen must be closed before you will see the modified record.

6. Click **Close** when finished.

Delete a Subscriber

It is expected that it would rarely be necessary to manually delete a subscriber as subscriptions automatically lapse after a period of twelve months when the subscriber data is automatically deleted. However, subscriber data can be manually deleted.

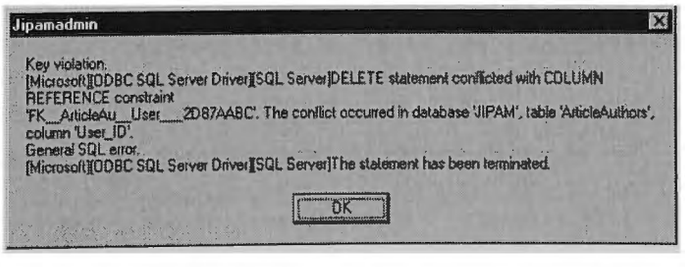
➢ To delete a subscriber

1. Click **View Subscribers** on **Main Screen JIPAM Admin. System**.

2. **Select** the Subscriber on the **View Subscribers** screen.

3. Click **Delete**. (There will be no confirmation request).

4. If the subscriber cannot be deleted, an error message box (similar to below) will be displayed and the record will not be deleted.



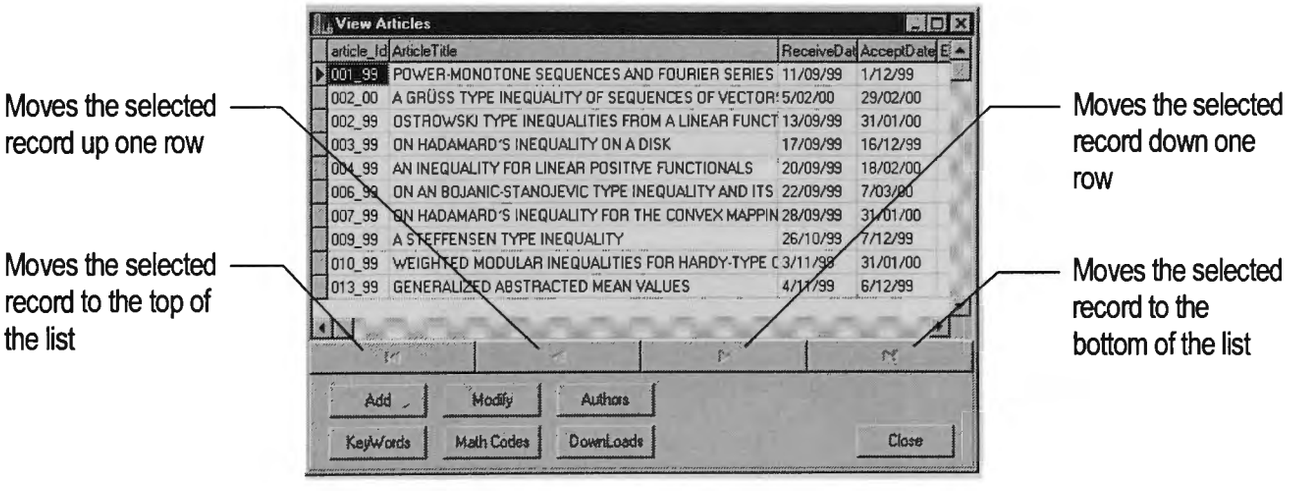
Chapter 4

Articles

THe Admin. Program provides the administrator with the facility to manage the articles published in JIPAM. Articles can be added to the database as soon as the draft article is received from the author. These drafts are then immediately available online (as new papers). When they are accepted for publication, their status changes and they are included in the contents of the current volume/issue.

Articles comprise two components: (1) the article details which includes the article title, author/s, keyword/s, maths code/s and the abstract (which may contain image files if formulas are discussed in the abstract) and (2) the full text of the article.

Click **View Articles** on **Main Screen JIPAM Admin. System**. The **View Articles** screen provides easy access to the articles.



The dialogue box displays all the articles in the database in tabular form and is resizable to allow more or less detail to be shown in the table. You can also adjust the width of individual columns using the cursor in the title bar of the table. However, your column size changes are not saved and the dialog box will revert to the default column sizes (see above) the next time you select **View Articles**.

Four navigation buttons are also provided to allow the position of the selected article to be changed in the database table (as shown above).

Adding and maintaining articles is structured around the following six actions.

- Add an article
- Modify an article
- Add/Modify an article's author
- Add/Modify an article's keywords
- Add/Modify an article's maths code
- Add/Modify an article's PDF files

Add an Article

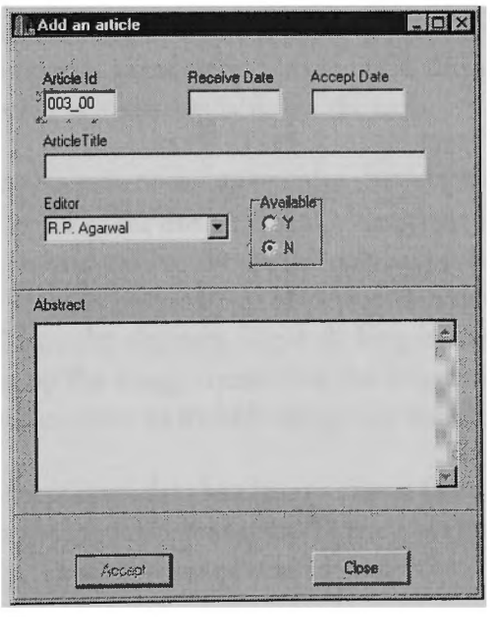
When articles are first added to the database, they are flagged as **Unpublished Articles**. After they have been published in a volume/issue, their status is changed to **Published Articles** (See Chapter 5 for more information). There are several steps to adding an article.

➢ To add an article

1. Click **View Articles** on **Main Screen JIPAM Admin. System**.

2. Click **Add** on the **View Articles** screen.

3. The **Add an article** dialog box is displayed. It is resizable to allow the abstract text to be entered into the abstract memo box.



4. Fill in the required **data fields** (detailed explanations of the data fields are provided below).

- *Article Id (Identifier)* is pre-filled from the database with the next article id sequence. The value can be changed but it must be of the form '999_99' where 999 represents the article number and 99 represents the year the article was received.

- *Article Title* is where the text of the new title is entered. It is necessary to enter at least one non-space, printable character.

- *Receive Date* is the date the article draft was received. A valid date must be entered into this field; otherwise, an error message will be displayed.

- *Accept Date* is the date the article was accepted (approved) for publication. This field may be left blank but if a date is entered it must be valid otherwise an error message will be displayed.

- *Editor* contains a drop down list of the journal's editors. The editor accepting the article is chosen from this list. New editors can be added to the list by changing the user role as described in the 'Modifying a Subscriber' section in the previous chapter.

- *Available* provides two options to determine if the article being added to the database is to be available for web-based activities such as searching, or author article listings. Click **Y** (yes) if the draft article is to be made available online. The default for a new article is set to **N** (no). This flag is provided to allow the administrator to enter all the article details before it is made available on the web.

- *Abstract* is where the text of the abstract is entered. Only text can be added in this field that means that HTML code can also be entered to control the display of the article on the web. This is necessary to incorporate functions (as image files) into the abstract. Right clicking in the *abstract* area will bring up the image menu. See the following section for more details on how to include images in the abstract.

5. Click **Add** when all necessary data has been entered (**Add** will only be enabled when both the *Article Title* and the *Receive Date* have been completed correctly. This adds the new article's details into the database and closes the dialog box.

Note

If the article details entered cannot be written to the database, an error message will be displayed and the dialog box will not close.

6. Click **Close** to leave without adding any new articles to the database.

Add Images to an Abstract

Some abstracts may need to contain images because mathematical formula may also be referenced in the abstract text. As the abstract field in the database is a text field, two different methods can be used to include image files into the abstract.

Hard code the <IMG> HTML tags in the abstract field and point the SRC attribute (source) to a hard coded file name.

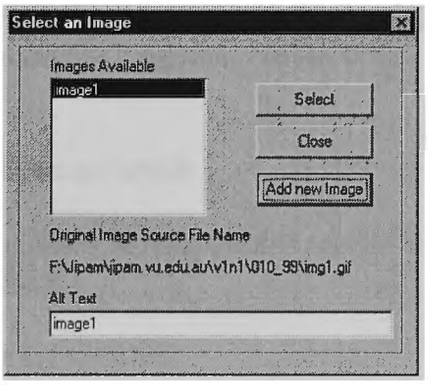
Use a marker scheme developed for the Admin Program that stores the image file in the database and references the image in the abstract. Using the marker scheme, an image is identified in the abstract by the code sequence **<##imagenn#alt-text##>**. The **nn** sequence identifies the image number and the **alt-text** sequence is the alternate text that will appear in the HTML response. In the web process, the ASP page code examines the abstract text looking for the markers and replaces the marker text with the appropriate HTML code to display the correct image.

➢ To include images into an abstract using the above marker scheme

1. Right click in the *abstract* area in either the **Add An Article** or **Modify An Article** screen to display a menu with the following two options.

    - *Insert Image* that will be enabled if the cursor position is not inside an existing image marker in the abstract.
    - *Change Image* that will be enabled when the cursor is inside an existing image marker.

2. Click either **Insert Image** or **Change Image** to display the **Select an Image** dialog box.



3. **Enter** the appropriate data (more detailed information follows) to either add or change image files.

- The *Images Available* list box displays a list of the images that have been previously been assigned to the currently selected abstract. Click **Add New Image** to include additional entries.

- The file name and path shown below the *Original Image Source File Name* label records the location from which the selected image file was originally sourced when it was added to the abstract. The file is read into the database when the abstract is saved.

- The *Alt text* contains the alternate text for the image. There is a limit of 255 characters.

4. Click **Select** to write the currently selected image in the images available list box into the abstract using the marker scheme. If the menu selection was **Change Image** the existing marker in the abstract is removed and the marker for the selected image is inserted into the abstract.

5. Click **Close** to exit the **Select the Image** dialog box. Changes made while the dialog box was open remain in effect until the abstract is saved.

Note

Images can be used in more than one location in the one abstract.

➢ To remove an image from an abstract

An image in an abstract can also be removed by deleting the marker text from within the text entered in the **Abstract** field on the **Add an article** dialog box.

Modify an Article

All of the article details originally added into the database (See the previous **Add an article** section) could be modified using this option.

➢ To modify the details in an article

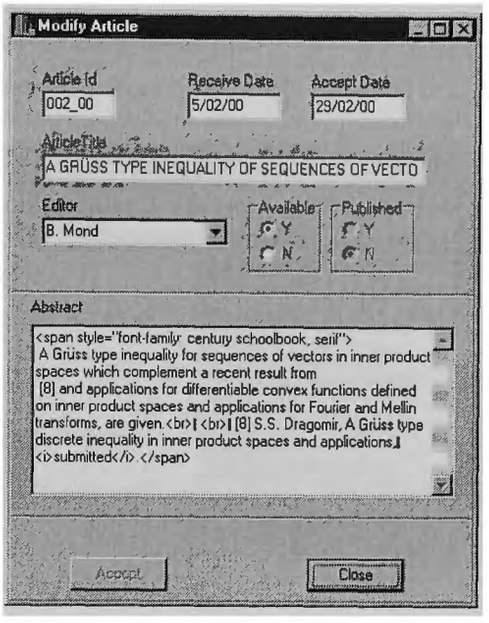
1. Click **View Articles** on **Main Screen JIPAM Admin. System**. Select the required article from the list.

2. Click **Modify** on the **View Articles** screen to display the **Modify Article** dialog box that shows the selected article's data.

3. Modify the **data fields** as required.

Note

Comments regarding the field contents as described in the previous section on adding new articles also apply to modifying article data.



- The **Available** option box is enabled only if the article has not been published thus allowing an article to be made available in the Web environment before it is published.

- The **Published** option box is a read-only field that indicates whether this article currently appears in a JIPAM volume/issue. It is a system flag that changes automatically from N (no) to **Y** (yes) when the article is added to the table of contents of the volume/issue in the process of being publishing.

4. Click **Update** to save the modified data. At this point, the article record in the database is changed.

5. Click **Close** when finished.

Add/Delete/Reorder Article Authors

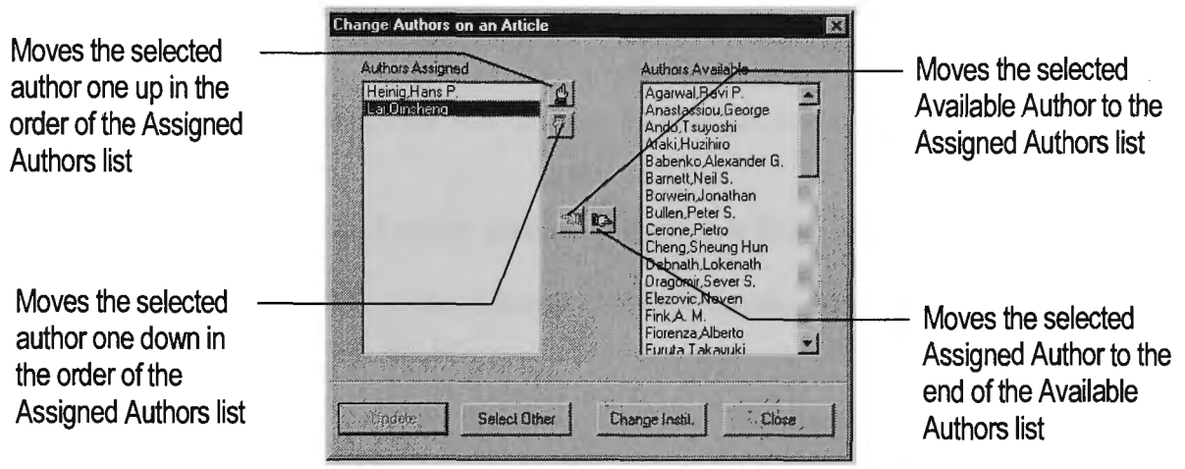
As part of the process to add a new article, it is necessary to assign authors to it. It is also possible that author names may have to be deleted from a specific article or the order of authors may need to be revised.

➢ To add, delete or reorder authors of an article

1. Click **View Articles** on **Main Screen - JIPAM Admin. System**. Select the required article from the list.

2. Click **Authors** in the **View Articles** screen to display the **Change Author of an Article** dialog box.

3. The **Change Author of an Article** dialog box contains two list boxes: (1) authors assigned and (2) authors available.



- The *Authors Assigned* list box on the left of the screen shows the authors already assigned to the article. This list box also shows the order of the authors, as they appear when the article is displayed.

- The *Authors Available* box lists all the subscribers who are currently classified as authors and who are not assigned to the current article. This list box is sorted on author surname order.

4. Click the relevant **Hand** buttons to make the changes to the author/s assigned to the selected article as required.

5. Click **Update** to save the changes.

6. Click **Close** to leave the dialog box without saving any of the changes.

Note

Select Other. It could be possible that the author of the selected article may not appear in the *Authors Available* list. However, it is likely the author is already entered into the database as a subscriber (that is, their currently assigned role is General). In this case, click **Select**

Other and then refer to the instructions in the next section, *Changing a Subscriber Role to an Author Role*

Change Instit (Institution). This button allows the institution of the currently assigned author to be quickly changed (bypassing the **View Users** screen. Refer to the section titled *Change an Article Author's Institution s* later in this chapter.

Change a Subscriber Role to an Author Role

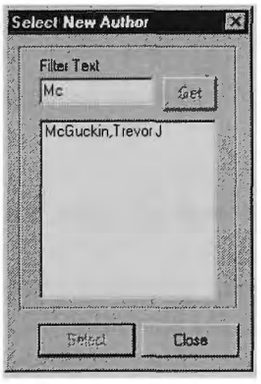
In assigning an author to a specific article, a situation could arise where the article's author is not yet flagged as an author (i.e. their name will not appear in the **Authors Available** list box. Thus, their role needs to be changed from subscriber to author.

➢ **To enter a new author (by changing their role from subscriber to author)**

1. Click **View Articles** on **Main Screen - JIPAM Admin. System**. Select the required article from the list.

2. Click **Authors** in the **View Articles** screen to display the **Change Author of an Article** dialog box.

3. Click **Select Other** to display the **Select New Author** dialog box which lists all subscribers not already classified as authors. A filter capability has been provided to streamline the process of locating the relevant subscriber's record in the database. It is not necessary to enter the full surname, just enter part of the subscriber's surname.



4. In the Filter Text box, enter the text you want to filter on. After at least one character has been entered, the **Get** button is enabled.

5. Click **Get** to display the subscriber surnames that match the filter text.

6. Click the required **Name** from the list to enable the **Select** button.

7. Click **Select** to write the selected subscriber into the *Authors Assigned* list box on the **Change Author of an Article** dialog box. At this point, the

subscriber's record in the database is changed (e.g. their role is changed from subscriber to author) and the **Add New Author** dialog box is automatically closed.

8. Click **Close** to leave the **Add New Author** dialog box without saving any changes.

Change an Article Author's Institution

After an author's name appears in the *Authors Assigned* list box associated with a specific article, their current institution details are automatically added to the article details. It is possible that the database may not contain their correct institution details (e.g. they may have changed institutions after they subscribed without updating their subscriber details).

Note

In regard to the author's institution details, the institution assigned to an article/author combination is maintained separately from the author/institution combination. This recognises that authors may change institutions during their academic life but once an article is published, the author/institution combination should not be changed to reflect such moves between different institutions.

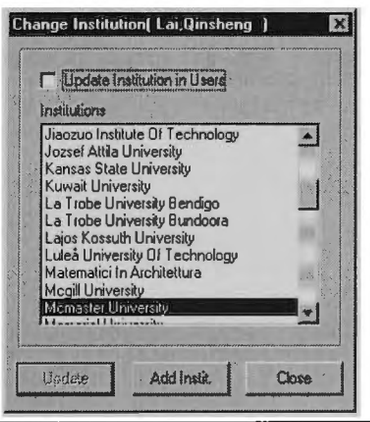
➢ To change the institution details of an author assigned to an article

1. Click **View Articles** on **Main Screen - JIPAM Admin. System**. Select the required article from the list.

2. Click **Authors** in the **View Articles** screen to display the **Change Author of an Article** dialog box.

3. In the *Authors Assigned* list box, select the **Specific Author** to be changed.

4. Click **Change Institution** button to display the **Change Institution** dialog box (the selected author's name will also appear in the title bar). A list of all the institutions currently recorded in the database is displayed with the author's existing institution highlighted.



5. Select the **new (and more recent and correct) institution** from the list which then enables the **Update** button.

Note

If the new institution is not included in the institution list that appears in the **Change Institution** dialog box, click **add Instit.** to open the **Add Institution** dialog box.

For more details on adding a new institution, refer to chapter 6.

6. Select the check box **Update Institution in Users** to enable the selected institution to be recorded as the current institution in the subscribers record within the database when the **Update** button is clicked.

7. Click **Update**. If no problems exist, the screen will close and the **Change Author of an Article** dialog box becomes the active window.

8. Click **Update** on the **Change Author of an Article** dialog box to save the new institution details to the database.

9. Click **Close** at any stage to leave the **Change Institution** dialog box or the **Change Author of an Article** dialog box without making any changes to the database.

Add/Modify the Keywords of an Article

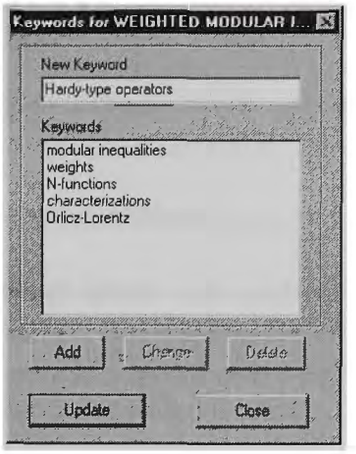
Keywords need to be entered into the database as part of the process to add a new article. Existing keywords may also need to be modified or deleted

➢ To add or modify an article's keywords

1. Click **View Articles** on **Main Screen JIPAM Admin. System**. Select the required article that you want to add or modify the associated keywords.

2. Click **Keywords** on the **View Articles** screen.

3. The **Keywords for <title of article selected>** dialog box is displayed with a **Keywords** list box that shows the keywords currently applicable to the selected article. There is also a **New Keyword** edit box that allows keywords to be added or modified in the dialog box.



4. To enter a new keyword, click **Enter** in the **New Keyword** edit box. The **Add** button is enabled after at least one character has been entered

5. To change an existing keyword, select it in the **Keywords** list box and then click **Change** to move it into the **New Keyword** edit box where it can be modified.

6. Click **Add** to move the keyword entered (either a new or a changed keyword) in the **New Keyword** edit box into the **Keywords** list box. **Add** also clears the **New Keyword** edit box.

Note

If a mistake is made in moving an entry from the **Keywords** list box into the **New Keyword** edit box, clicking **Add** will move it back into the **Keywords** list box.

7. Click **Delete** to remove the keyword selected in the **Keywords** list box.

8. Click **Update** when all necessary changes to the **Keywords** list box have been done. This adds the changes to the database and closes the dialog box. (The **Update** button will only be enabled if changes have been made to the entries in the **Keywords** list box.).

9. Click **Close** to leave the **Keywords for <title of article selected>** dialog box without saving any changes.

Add/Modify the Math Code of an Article

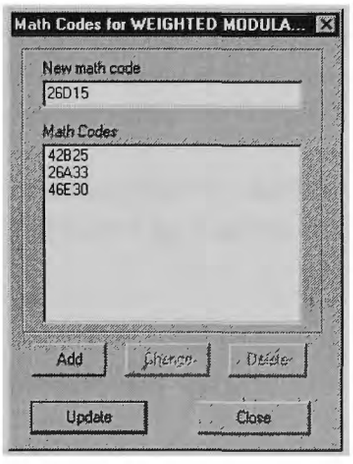
Math codes need to be entered into the database as part of the process to add a new article. Existing math codes may also need to be modified or deleted

➢ To add or modify an article's math codes

1. Click **View Articles** on **Main Screen JIPAM Admin. System**. Select the required article that you want to add or modify the associated math codes.

2. Click **Math Codes** on the **View Articles** screen.

3. The **Math Codes for <title of article selected>** dialog box is displayed. It contains a **Math Codes** list box that displays the math codes currently applicable to the selected article. There is also a **New math code** edit box that allows new math codes to be added or existing maths codes to be modified.



4. To enter a new math code, click in the **New math code** edit box. The **Add** button is enabled after at least one character has been entered.

5. To change an existing math code, select it in the **Math Codes** list box and then click **Change** to move it into the **New math code** edit box where it can be modified.

6. Click **Add** to move the math code entered in the **New math code** edit box (which could be either a new or a changed math code) into the **Math Codes** list box. **Add** also clears the **New math code** edit box.

> **Note**
>
> If an entry is moved from the **Math Codes** list box into the **New Math Code** edit box in error, clicking **Add** will move it back into the **Math Codes** list box.

7. Click **Delete** to remove the math code selected in the **Math Codes** list box.

8. Click **Update** when all necessary changes to the **Math Codes** list box have been done. This adds the changes to the database and closes the dialog

box. (The **Update** button will only be enabled if changes have been made to the entries in the **Math Codes** list box.).

9. Click **Close** to leave the **Math Codes for <title of article selected>** dialog box without saving any changes.

Add/Modify the PDF Files for an Article

PDF files of the article full text need to be entered into the database as part of the process to add a new article. Once a PDF file has been written to the database, it can only be updated by deleting the existing entry and the adding in the new PDF file.

In Phase 1, there are only two types of PDF files supported, e.g. PDF screen and PDF printer. However, future enhancements to increase the number of file types are possible by adding additional entries in the PrintFileTypes table in the database.

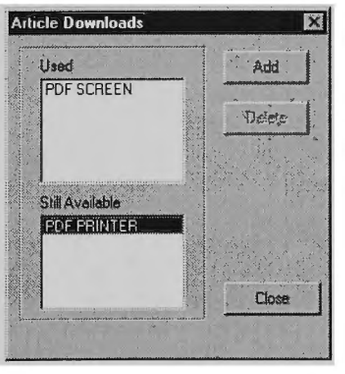
Since this is not expected to occur very often, a facility has not been provided in Phase 1. However, it could be achieved by a simple insert statement at some future date.

➢ To add or modify an article's PDF files

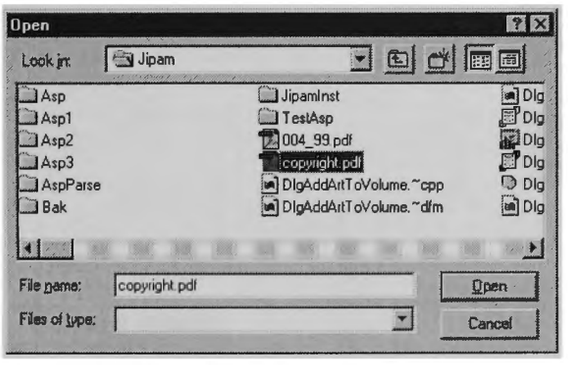
1. Click **View Articles** on **Main Screen JIPAM Admin. System**. Select the required article that you want to add or modify the associated keywords.

2. Click **Download** on the **View Articles** screen.

3. The **Article Downloads** dialog box shows two list boxes: **Used** and **Still Available**. If any PDF files are currently associated with the selected article, the type is displayed in the **Used** list box (e.g. it could be either the PDF screen or PDF printer version of the full text PDF file or both). The **Still Available** list box shows the types of file that are still able to be associated with the selected article.



4. To add a PDF file to the selected article, select the file type from the **Still Available** list and click **Add**. This opens a file search dialog box.



5. Select the file to be inserted into the database by navigating through the file structure and click **Open**. The file selection dialog box will close and you will be returned to the **Article Downloads** dialog box

6. A progress indicator bar showing the percentage of the transfer process completed appears at the bottom of the **Article Downloads** dialog box.

7. When the transfer is 100% complete, the progress indicator bar disappears and the type of PDF file type that has just been transferred moves from the **Still Available** list into the **Used** list.

8. To delete PDF files from the selected article, select the file type from the **Used** list and click **Delete** which moves it into the **Still Available** list.

9. Click **Close** to leave the dialog box.

Volumes/Issues

The Admin. Program provides the administrator with the facility to manage the article publication process. New volume/issues can be created and published on the JIPAM web site. Approved articles can be added to the Table of Contents of the new volume/issue. Additional options are also provided to allow the contents of each volume/issue of JIPAM to be modified.

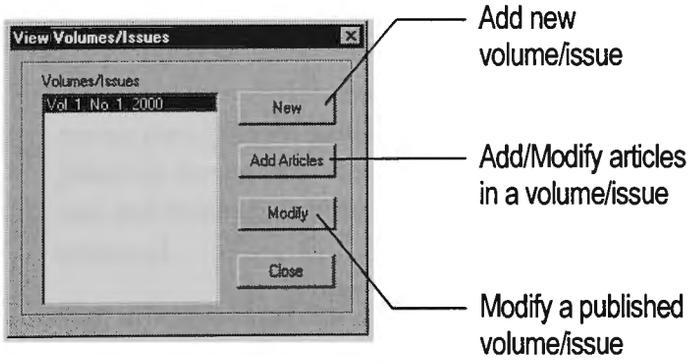
Note

Each JIPAM volume contains all the issues published in one calendar year. Currently, it is planned to publish two issues of JIPAM each year. In this guide, the terminology used to identify a specific issue of the journal is volume/issue.

Volume/Issue Maintenance

To maintain data related to the contents of specific issues of the journal, it is first necessary to view the volumes table in the database.

Click **View Volumes/Issues** on **Main Screen JIPAM Admin. System**. The **View Volumes/Issues** dialog box is displayed with a list box showing all of the volumes currently in the database. **Select** the relevant issue of the journal from the list.



Maintenance of volume/issue data is structured around the following three actions.

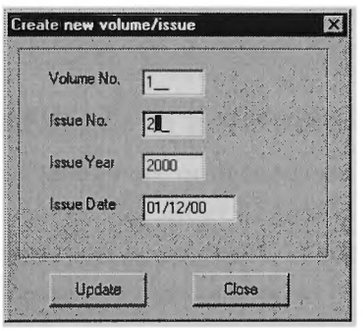
- **Add** (create) a new volume/issue displays a dialog box to enter the attributes of the new volume/issue being created.
- **Add** (modify/reorder) **articles** is used to add unpublished articles to the currently selected volume/issue. It also allows the publication order (the order that the articles will appear in the table of contents) within the currently selected volume/issue to be changed. Articles can also be moved between the published and unpublished article lists.
- **Modify** a volume/issue changes the attributes of the volume/issue currently selected.

Create a New Volume/Issue

New volumes/issues will need to be created on an ongoing basis.

➢ To create a new volume/issue

1. Click the **View Volumes** button on **Main Screen - JIPAM Admin. System**. Select the required volume/issue from the list.
2. Click **New** on the **JIPAM Volumes/Issues** screen to display the dialog box in which the new volume/issue details are entered.
3. Fill in the required **data fields** (detailed explanations of the data fields are provided below).



- *Volume No.* represents the publication year and is numbered consecutively from Year 2000 (Volume No.1). The field is preset to accept only numeric digits to a length of between one and three characters but the consecutive sequence is not enforced.
- *Issue No.* are numbered consecutively and identify the issues published in a specific year (a volume). The field is preset to accept only numeric digits to a length of between one and three characters. Again, the consequence sequence is not enforced.

- *Issue Year* identifies the year the volume represents. The field is preset to accept only numeric digits to a length of four characters and all four digits are required so you must enter the full year. The relationship with the Volume No. is also not enforced by the database.

- *Issue Date* is self-explanatory. The field is preset to accept only valid dates. Once the field is selected for input (has focus), no other field or button on the form can be selected until a valid date has been entered in this field. At this stage, the date of issue does not appear in any reports or on any display screens and is really for information only.

4. Click the **Update** button (which will become available if any of the above fields have been modified) to record the details entered in the database. If the addition of the new volume/issue details is successful, the dialog box will close.

Note

If there is a problem with any of the fields, the update will fail and an error message will be displayed.

5. Click the **Close** button to close the dialog box without making any changes to the database.

Add/Modify/Reorder Articles in a Volume/Issue

When approved, unpublished articles are published or added to the table of contents of the next volume/issue. The order of articles in the selected volume/issue's table of contents can be changed. In addition, there may be certain circumstances where it is necessary to change articles from being published to unpublished in a specific volume/issue.

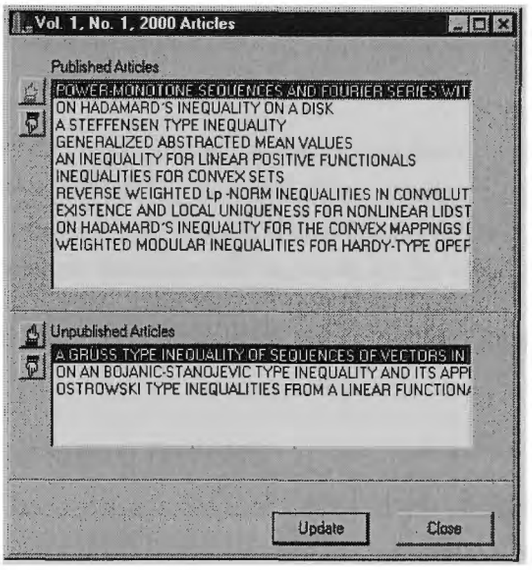
Note

The assumption is that once an article has moved from being 'unpublished' to 'published', it should remain 'published' (that is, within the table of contents of that volume/issue). However, in the case of an error being detected, an option has been included to allow articles to be added, removed or changed in any volume/issue.

➢ To add, modify, reorder articles in a volume/issue

1. Click **View Volumes/Issues** on **Main Screen - JIPAM Admin. System**. Select the required volume/issue from the list.

2. Click **Articles** in the **View Volumes/Issues** screen to display a resizable dialog box containing two list boxes: (1) articles published in the volume/issue selected and (2) articles currently unpublished. When first displayed, the **Published Articles** list box contains the article titles that are currently published to the volume/issue shown in the title bar. The **Unpublished Articles** list box shows articles that are currently unpublished and which do not yet appear within the table of contents of any volume/issue.

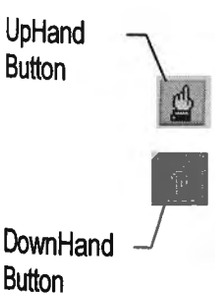


3. To **add** articles to the current volume/issue, select the title of the article from the **Articles Unpublished** list and **click** the **UpHand** button. This moves the article to the **Articles Published** list where it is flagged in the database as being published.

Note

There are two hand buttons with different functions on the left side of each list box. Their function is determined by which list they are next to.

- *Articles Published List.* The **UpHand** button and the **DownHand** buttons control the order of the articles in this list. Click the **UpHand** button to move the currently selected article up one place and click the **DownHand** button to move the currently selected article moved down one place in the **Articles Published** list box.



- *Articles Unpublished List.* The **UpHand** button and the **DownHand** button control the movement of articles from one list box to the other. By clicking the **UpHand** button, the currently selected article in the **Articles Unpublished** list is appended to the **Articles Published** list box (becoming part of the table of contents of the current volume/issue of JIPAM). If an error is made and the article should not have been published, it is possible to move a published article back into the **Articles Unpublished** list by clicking the **DownHand** button which flags the article as unpublished in the database. However, care must be taken to ensure that the online version of JIPAM remains consistent with the printed version in terms of the table of contents of each volume/issue. *Note that moving articles from one list to another changes the order of the articles in both lists.*

4. Click the **Update** button to save the changes you have made to the database and to close the dialog box. All changes are effected as a database transaction. If the transaction fails for whatever reason, an error message will be displayed and no changes will be made to the database.

5. Click **Close** to leave the dialog box without saving any of the changes.

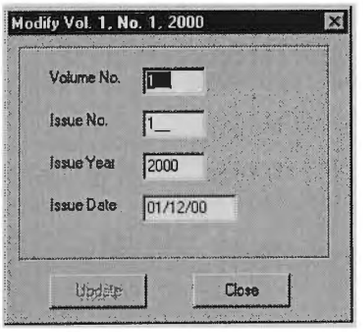
Change a Volume/Issue Details

Once a volume/issue has been created and issued, there should be no need to change any of these details. This option would only be used if an error were to be made when the original volume/issue was created.

➢ To change a volume/issue details

1. Click **View Volumes** on **Main Screen - JIPAM Admin. System**. Select the required volume/issue from the list.

2. Click **Modify** on the **JIPAM Volumes/Issues** screen to display the dialog box in which the new volume/issue details are entered.



3. **Enter** the data required in the input screen (this screen is identical to the **Add new Volume/Issue** screen discussed previously).

4. Click **Update** to modify the database with the amended details. If the volume/issue update is successful, the dialog box will close.

Note

If the volume/issue update is not successful, an error message will be displayed and the dialog box will remain open.

5. Click **Close** to close the dialog box without making any changes to the database.

Chapter 6

Institutions

Data is recorded in the database about the current institutions with whom JIPAM's subscribers (including authors and editors) are associated because there are many benefits in knowing the institutional background of the journal's readership base. The following two levels of institutional data are recorded.

- The **Affiliation** field will generally be completed online when users register as new subscribers. They can then modify the affiliation information online at any stage in the future.
- The **Institution** field can only be completed by the administrator using the Admin. System program.

This two level data structure have been implemented for the following reasons: (1) to address the lack of security associated with web based information, (2) to ensure a consistent approach to the display of the institution details associated with authors and editors and (3) to provide a way of maintaining current institutional information for authors or editors who move from one institution to another.

Default Settings for the Institution Field

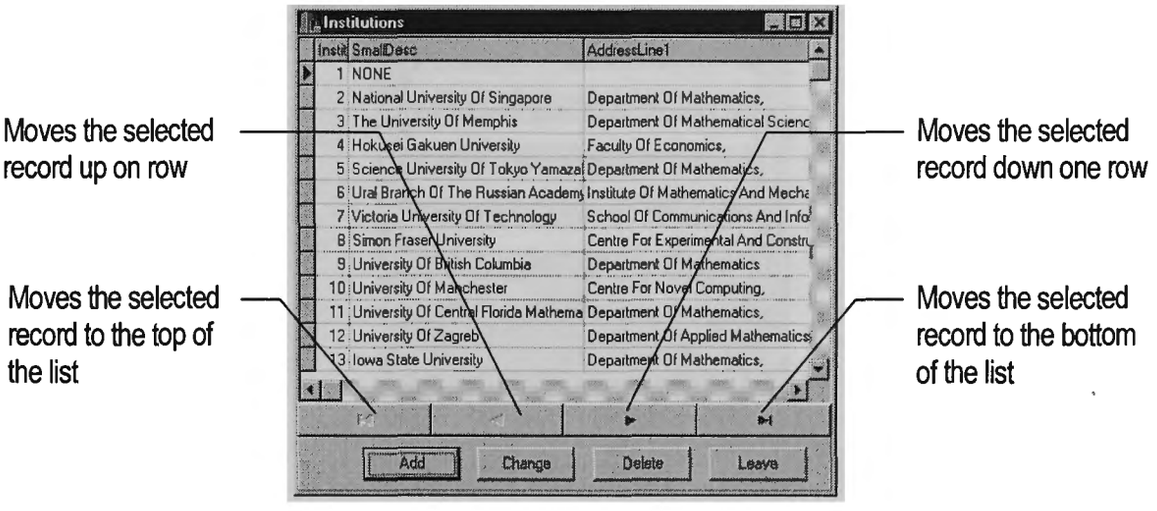
When the JIPAM database is originally created, an institution with **None** in the **Name** field is automatically added to the database as the default entry in the institution table. This default institution cannot be deleted or modified in the institution table of the database.

The default (**None**) is automatically entered into the record of all new subscribers. For those subscribers who are also authors and/or editors, the System Administrator needs to manually change **None** to the appropriate institution from the drop down list box. Refer to the *Modify a Subscriber* section in Chapter 3 Subscribers for more information about how to do this.

Institution Data Maintenance

To maintain the institutional data recorded, it is first necessary to view the institutions table in the database.

Click **View Institutions** on **Main Screen JIPAM Admin. System**. The **View Institutions** dialog box is displayed with a list box showing all of the institutions currently in the database and their associated address details in tabular form.



The dialogue box is resizable to allow more or less detail to be shown in the table. Four navigation buttons are provided to allow the position of the selected subscriber's record data to be changed in the database table (as detailed above).

Maintenance of data on institutions is structured around the following three actions.

- **Add** a new institution displays a dialog box for the entry of the institution's details.
- **Modify** allows the details of the institution that is currently selected to be changed.
- **Delete** removes the currently selected institution from the database.

Add a New Institution

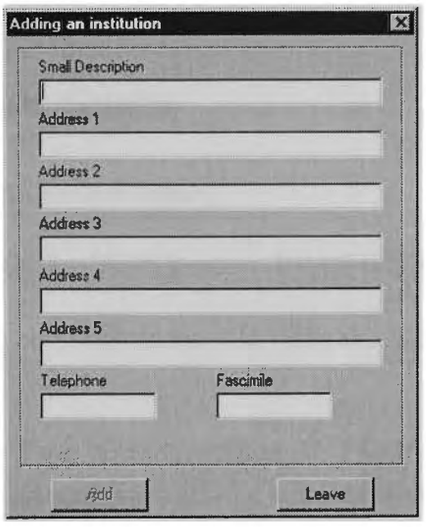
A new institution can only be added into the database if it is unique. Thus, it is important that the current institutions listed in the database are very thoroughly checked by the administrator to ensure the institution does not appear under a slightly altered name BEFORE a new institution is added.

One of the reasons for a two level approach to institutional data was to ensure that there is consistency in how the way that institutional data is displayed.

➢ To add a new institution

1. Click **View Institutions** on **Main Screen JIPAM Admin. System**.
2. Click **Add** on the **View Institutions** screen.

3. The **Add Institution** dialog box is displayed.



4. **Enter** the data required. The **Institution Name** is self-explanatory and is the only field that must be completed in this dialog box. Once the cursor is in the field, at least one character must be entered before any other field or button can be selected because the minimum size of this field is one character. The maximum sizes for all fields are shown below.

• Institution Name	60 characters
• Address 1	255 characters
• Address 2	255 characters
• Address 3	255 characters
• Address 4	255 characters
• Address 5	255 characters
• Telephone	60 characters
• Facsimile	60 characters

Note

All the above fields will accept any printable character and once any of the fields has been modified, the **Update** button becomes enabled. Any leading or trailing spaces in the text entered in any of the fields are discarded before they are written to the database.

5. Click **Update** to write the new institution details to the database. If no problems exist, the screen will close.

Error Message

If there is a problem with the **Institution Name** field, a message will be displayed.

6. Click **Close** to leave the Institutions table without adding any new institutions to the database.

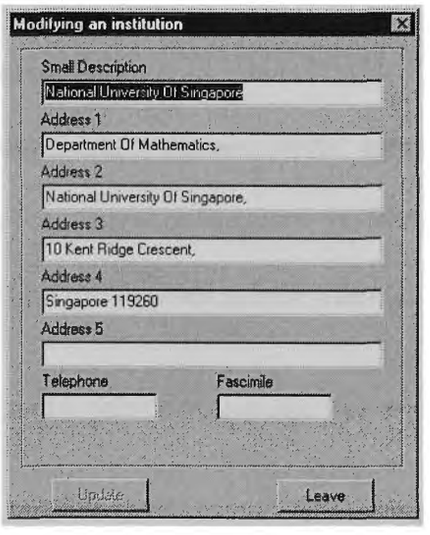
Modify an Institution Details

➢ **To modify an institution's details**

1. Click **View Institutions** on **Main Screen JIPAM Admin. System**.

2. Select the **institution** you wish to modify from the **View Institutions** screen.

3. Click **Change** on the View Institutions screen. The **Modify Institution** dialog box is displayed complete with the selected institution's data.



4. **Modify** the data as necessary. All of the fields (including the **Institution Name**) can be changed using this dialog box. Refer to the information provided in the previous section (Add a New Institution) for more detail about these fields.

Note

All the fields will accept any printable character and once any of the fields has been modified, the **Update** button becomes enabled. Any leading or trailing spaces in the text entered in any of the fields are discarded before they are written to the database.

5. Click **Update** to write the modified institution details to the database. If no problems exist, the screen will close.

Error Message

If there is a problem with the **Institution Name** field, a message will be displayed.

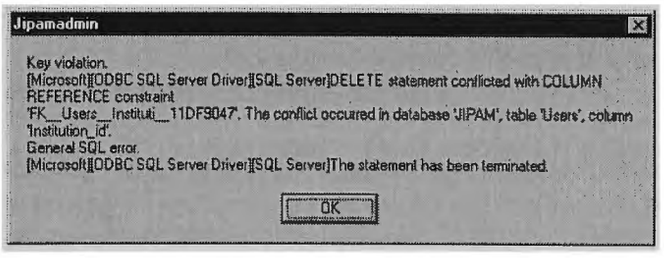
6. Click **Close** to leave the Institutions table without adding any new institutions to the database.

Delete an Institution

Any institution other than the default institution (that is, the entry with **None** in the **Institution Name** field) may be deleted provided the institution is not referenced anywhere else in the database.

➢ To delete an institution

1. Click **View Institutions** on **Main Screen JIPAM Admin. System**.
2. Select the **institution** you wish to delete from the **View Institutions** screen.
3. Click **Delete** on the View Institutions screen. There is no request to confirm the deletion.
4. If the institution cannot be deleted, a message box similar to that shown below will be displayed.



Chapter 7

Statistics

User traffic to the JIPAM web site is tracked. The first time in a session that a user accesses the site, statistical data is gathered from the HTML request header. The data recorded includes the browser type being used; the primary language set in that browser as well as some basic referral information.

User Tracking

Every time a user accesses the JIPAM web site, a new session object is created by the web server provided they do not already have a current session object. As part of the creation of the session object, a record is written to the Sessions database table. The code for writing the record is contained in the session's **on_start** procedure that resides in the **global.asa** file in the root JIPAM directory.

The record written to the database contains information extracted from the users' request header and it is raw data. The raw data needs to be summarised into other tables before it can be analysed and reported on. The following two database procedures are provided to support this summarisation process.

SummariseSession.sql which summaries the data

DeleteOldRecords.sql removes the summarised records from the Sessions table as well as removing all records over 26 weeks old from the other statistical tables in the database.

As part of the original database setup procedure (See Chapter 1), an SQL job called **Weekly JIPAM Session Update2** is created which then runs these two procedures once a week. However, the job will only run if the SQL Server Agent is active in the database (refer to your database administrator).

Note

If, for whatever reason, the procedure is not run for a week or more, the code within the SQL procedures will automatically take care of summarising the weeks that have been missed. This means that the summarisation process could be run every four weeks instead of every week, if that is preferable.

Visitor Reports

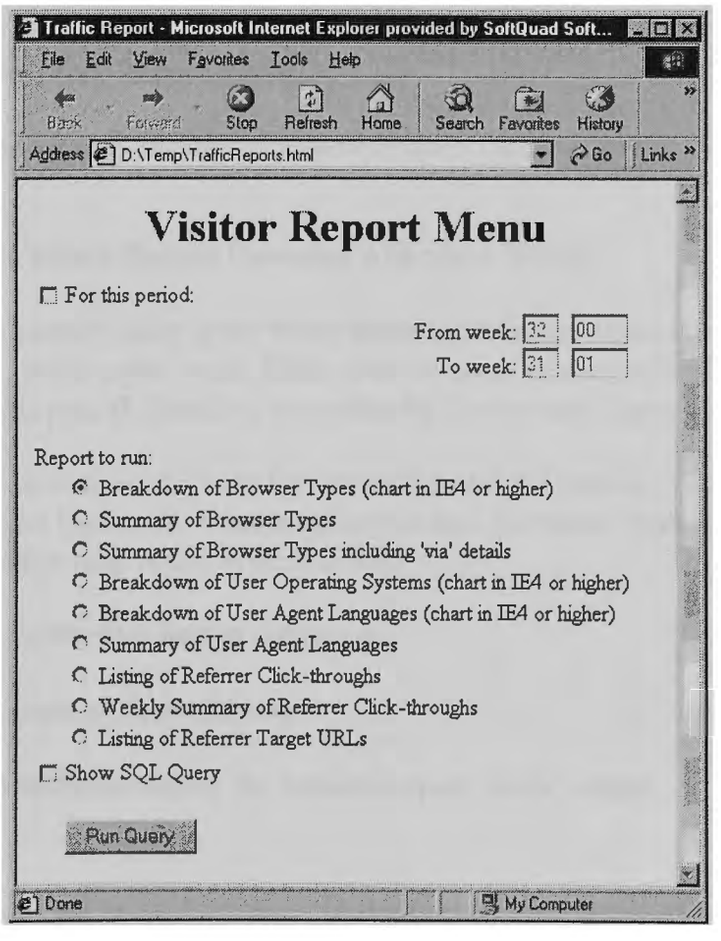
Different visitor reports can be created as an aid to analysing site usage. The nine reports included in Phase 1 of the JIPAM Web System are as listed below.

- Breakdown of Browser Types (chart in IE4 or higher)
- Summary of Browser Types
- Summary of Browser Types including 'via' details
- Breakdown of User Operating Systems (chart in IE4 or higher)
- Breakdown of User Agent Languages (chart in IE4 or higher)
- Summary of User Agent Languages
- Listing of Referrer Click-throughs
- Weekly Summary of Referrer Click-throughs
- Listing of Referrer Target URLs

➢ To Run a Visitor Report

First, it is necessary to login to the JIPAM web site as administrator. The REPORTS menu button appears on the bottom of the menu sidebar. Click this button to display the **Visitor Report Menu**.

Click the **REPORTS** button from the JIPAM sidebar menu



Note

Two separate time frames can be chosen for the visitor's reports. The first specifies an individual week while the second reports on a range of weeks with a starting week and an ending week. The 'For this Period' checkbox determines which report is run. If the checkbox is selected, the report covers the range of weeks entered into the two input boxes. However, if the checkbox is not selected, the report generated covers the week specified in the 'From week' input boxes.

➢ To Run a Visitor Report Covering a Range of Weeks

1. Click on the **For this period** checkbox to enable the **From week** and **To week** input boxes. When the pairs of boxes initially appear, they contain the current week and year.

2. In the **From week** input area, enter the week that starts the period you wish to report on into the first input box and the associated year in the second input box.

3. In the **To week** input area, enter the week that ends the period you wish to report on into the first input box and the associated year in the second input box. Again, there are no validity checks undertaken on the values entered into these two input boxes.

4. Click the required **Report Name**.

5. Click **Run Query** to display the required report on the screen.

➢ To Run a Visitor Report Covering a Specific Week

1. Check the default value of the **From week** input boxes. If the specific week that you want to report on is different to the week currently shown, select the 'For this period' checkbox to enable the 'From week' input boxes.

2. Enter the starting week in the first input box and the starting year in the second input box of the **From week** input area. No error checking is attempted for valid values in these fields.

3. Deselect the **For this period** checkbox.

4. Click the required **Report Name**.

5. Click **Run Query** to display the required report on the screen.

Note

It is possible to view the SQL query that is constructed to run the required Visitor's Report. To do so, click the 'Show SQL Query' checkbox. You should note that it is not possible to view the SQL query for all the report types particularly the graphical reports as there are many queries constructed and run to support the generation of the report.

Glossary

ACCEPTING EDITOR'S NAME	The name of the JIPAM editor who accepts an article draft and ensures that the article is ready for publication.
AFFILIATION	The name of the establishment with which a subscriber is associated. This name assists the administrator when they associate an institution with an author or an editor.
ANNUAL DATE	Registration as a subscriber to JIPAM is for a defined period (12 calendar months from the date of registration). The annual date is the last day of the registration period.
ARTICLE	Journal article submitted in draft form which is reviewed and either (1) rejected or (2) accepted with revisions (known as unpublished articles) or (3) approved for publication in the next volume/issue of JIPAM with no changes (known as published articles).
ARTICLE ABSTRACT	An abstract (a summary of the specific article) is included at the start of every article published in JIPAM.
ARTICLE AUTHOR DETAILS	The name and contact details (a hyperlink to email and personal web site addresses) of the author/s is included on the first page of every article published in JIPAM.
ARTICLE AUTHOR INSTITUTION'S DETAILS	The name and contact details (a hyperlink to email and web site addresses) of the institutions of each author/s is included on the first page of every JIPAM article.
ARTICLE DETAILS	Provides the following details for each article: ▯ Article Title ▯ Article Author Details ▯ Author's Institutions ▯ Date Article Draft Received ▯ Data Article Draft Accepted for Publication ▯ Accepting Editor's Name ▯ Abstract ▯ Keyword/s ▯ Classification Code/s

ARTICLE ID	The unique character sequence used to identify an article with the form '999_99' (999 represents the article number and 99 represents the year the article was received).
ARTICLE TITLE	The article title is included on the first page of every article published in JIPAM and is assigned to the article by the authors.
ARTICLES PUBLISHED	Journal articles which have been approved and published.
AVAILABLE	The flag that determines whether details of an article in the JIPAM database is accessible from the web.
CLASSIFICATION CODE	Individual article authors provide a list of mathematical classification codes to help users/subscribers search JIPAM volumes for specific articles.
CURRENT SESSION	A session becomes current for a subscriber from the moment they access their first JIPAM web page. The session ends either when their web browser is closed down or 20 minutes has expired since they accessed a web page from JIPAM.
DATE ARTICLE DRAFT RECEIVED	The date that an article is received in draft form from the author.
DATE ARTICLE DRAFT ACCEPTED FOR PUBLICATION	The date that a JIPAM editor determines that an article draft is ready for publication.
DISPLAY NAME	The name an author or an editor would like associated with their articles when those articles are displayed on the web.
GENERAL USER	Any member of the WWW community whose browser is directed towards the JIPAM web site (URL xxxx). Users have access to the Table of Contents for all currently published volumes of the JIPAM complete with article abstracts. However, users are unable to access the full text of articles until they register as subscribers.
INSTITUTION	The formal name of the establishment an author or editor is associated with when submitting or editing an article for publication. Only the administrator can create institutions (See affiliation).
JOURNALS UNPUBLISHED	Journal articles that have been accepted but which are not yet approved for publication. Unpublished articles can be viewed from the **NEW PAPERS** button on the web site

KEYWORD	Individual article authors provide a list of keywords for users/subscribers to search JIPAM for specific articles.
LOG IN	To gain access to the full text of articles published in JIPAM, it is necessary to be registered as a subscriber and to be logged in.
PASSWORD	A subscriber's secret word for advising the system that they are indeed the subscriber who is logging in.
PUBLISHED	The internal JIPAM database flag that indicates that an article has been associated with a JIPAM volume/issue.
RENEWAL DATE	The date on which a subscriber's annual subscription expires. In phase 1, this date is automatically extended for the subscription period.
ROLE	The actions a subscriber can perform in the JIPAM system. All subscribers are assigned the General tag when they register online. The Author role is assigned to a subscriber who has submitted an article. The Editor role is assigned to a subscriber who is available to edit articles, while the Both tag is assigned to subscribers who are both an author and an editor.
START DATE	The date a user first registered as a subscriber with the JIPAM system.
SUBSCRIBER	A user of JIPAM who has completed the subscription form and had their details stored in the JIPAM database. A subscriber is allowed access to the full text of all articles published in JIPAM. This access allows download of each article file in PDF format.
SUBSCRIPTION	Online process to register users so that they can access and download the full text of all JIPAM articles. Currently, subscription and thus access to full text articles is free but charges may be levered at some stage in the future.
SUBSCRIPTION PERIOD	Registration as a subscriber of JIPAM is for a fixed period of 12 calendar months from the initial subscription date.
WEB ADDRESS	The URL address of a subscriber's home page.
VOLUME	A JIPAM volume contains all the issues published in one calendar year. Currently, it is planned to publish two issues of JIPAM each year.