

Faculty of Computer Science & Information Technology

University of Malaya

Electronic Document Delivery System (EDDS)

Name	: Ng Eng Seong
Matric Number	: WEK 990230
Supervisor	: Pn Abrizah Abdullah
Moderator	: Pn Nor Edzan Haji Nasir
Session	: 2001/2002

This thesis is submitted to Faculty of Computer Science and Information Technology University of Malaya in partial fulfillment of the requirement for the Bachelor of Computer Science Degree.

Disclaimer

The ownership of this report is reserved by University of Malaya and no part of this report should be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, and recording or otherwise without prior written consent from University of Malaya.

Abstract

The Electronic Document Delivery System (EDDS) is a web-based system that promises a quick and efficient way to access information for University of Malaya. The entire request can be made electronically and the materials shall be sent to the requester electronically. The service is also available 24 hours per day throughout the year because it is available online. The EDDS will cater the request for an article of a journal or a chapter of a book, which is not found in the University of Malaya Library collections.

The objectives of the EDDS are to create an automated document delivery system for the University of Malaya Library, a user interface to help user navigate through the requesting stage in order to get their documents, an electronic form to replace the current hardcopy form, a interface for two types of request which are an article of a journal and a chapter of a book, and to developed a database to store user's information captured through electronic forms.

The Waterfall Methodology with Prototyping is used to develop the EDDS. EDDS is an active web page solution that needs special development tools and languages such as ASP, HTML, Internet Information Server (IIS), Visual InterDev, VB Script, Java Script and Front Page 2000.

The EDDS is implemented by developing the EDDS module by module and finally combined all the modules to create the whole EDDS system. After the EDDS system completes, the system testing is tested using internal testing, unit testing and application testing.

Few end users evaluated the EDDS, and the EDDS strengths and constraints are identified and some of the constraints are suggested as future enhancements. Throughout the EDDS development process, a lot of experience and knowledge was gained in web based system development.

The EDDS has meet all its objectives and requirements stated in the design documentation. It is hope that it will ease the effort to obtain information and at the same time promote more research work to be done in University of Malaya. It is also hoped that EDDS will contribute the Library of University of Malaya to provide more efficient services and help the librarian to plan for future services in document delivery services. The significances of the EDDS are to be seen, when it is implemented in the real working environment.

Acknowledgements

The development of this EDDS project was carried out along with advices, assistance, contributions and ideas from many individual

First and foremost, I would like to extend my most sincere gratitude to Pn. Abrizah Abdulah, my project supervisor who has provided me with unlimited support, time and guidance.

Next, I would like to express my special thanks to Pn. Hidayatun Nisak, my project sponsor, for her support and guidance. Special thanks also goes to the staff from the Faculty of Computer Science and Information Technology, University of Malaya for their cooperation and help in the development of EDDS.

Finally, I would like to thank all my friends and seniors especially Ng Yuet Woon for sharing their time and knowledge with me. Your supports and motivations are deeply appreciated.

Acknowledgements

The development of this EDDS project was carried out along with advices, assistance, contributions and ideas from many individual.

First and foremost, I would like to extend my most sincere gratitude to Pn. Abrizah Abdullah, my project supervisor who has provided me with unlimited support, time and guidance.

Not the least, my special thanks to Pn. Nor Edzan Haji Nasir, my project moderator for spending precious time to moderate this project and giving a lot of ideas to make this project more successful.

Special thanks also goes to the staff from the Faculty of Computer Science and Information Technology, University of Malaya for their co-operation and help in the development EDDS.

Finally, I would like to thank all my friends and seniors especially Ng Yuet Wean for sharing their time and knowledge with me. Your supports and motivations are deeply appreciated.

Table of Contents

Abstract	i
Acknowledgements	iii
Table of Contents	iv
List of Figures	x
List of Tables	xi
Chapter 1 Introduction	
1.1 Introduction	1
1.2 What is Electronic Document Delivery System	2
1.3 The Aim	2
1.4 Project Objective	3
1.5 Project Scope	4
1.6 Project Significance	5
1.7 Hardware and Software Requirements	6
1.8 Project Schedule	7
1.9 Definition	9
1.10 Summary	10
Chapter 2 Literature Review	
2.1 Importance of Literature Review	11
2.2 Electronic Delivery Services Review	12
2.2.1 Electronic Document Delivery System	13
2.2.2 Second Generation: Supply Chain Image Based Systems	16
2.2.3 Third Generation: Scalable Stand Alone Systems	17
2.6 Electronic Document Delivery System in Malaysia	18
2.7 Current Available Systems	19
2.8 Types of Information Available on Web Pages About	20
1. University of Illinois Libraries	23
2. Syracuse University Library	26
3. Stanford University Library	29
4. Arizona State University Libraries	32
5. RMIT University Library	35
6. Monash University Library	38
7. SHPL	41
8. MLSC	43
9. National Library of Australia	45
10. The Clement C. Maxwell Library	47
2.9 Summary of Web Sites Reviewed	49
2.10 Sample of Request Forms	52
2.11 Summary	58

Table of Contents

Abstract	i
Acknowledgments	iii
Table of Contents	iv
List of Figures	x
List of Tables	xi
<u>Chapter 1 Introduction</u>	
1.1 Introduction	1
1.2 What is Electronic Document Delivery Systems?	2
1.3 The Aim	2
1.4 Project Objectives	3
1.5 Project Scopes	4
1.6 Project Significances	5
1.7 Hardware and Software Requirements	6
1.8 Project Schedule	7
1.9 Definition	9
1.10 Summary	10
<u>Chapter 2 Literature Review</u>	
2.1 Importance of Literature Review	11
2.2 Document Delivery Services Review	12
2.3 Electronic Document Delivery Services Review	13
2.4 Benefits of Using Electronic Document Delivery Services (EDDS)	14
2.5 Evolution of Electronic Document Delivery Systems	15
2.5.1 First Generation: Online Ordering	15
2.5.2 Second Generation: Supply Driven Image Based Systems	16
2.5.3 Third Generation: Scalable Stand Alone Systems	17
2.6 Electronic Document Delivery System in Malaysia	18
2.7 Current Available Services	19
2.8 Types of Information Available on Web Pages About	20
1. University of Nevada Libraries	23
2. Syracuse University Library	26
3. Stanford University Library	29
4. Arizona State University Libraries	32
5. RMIT University Library	35
6. Monash University Library	38
7. SHPL	41
8. MUSC	43
9. National Library of Australia	45
10. The Clement C. Maxwell Library	47
2.9 Summary of Web Sites Reviewed	49
2.10 Sample of Request Forms	53
2.11 Summary	58

Chapter 3 Methodology and System Analysis

3.1 Methodology	60
3.1.1 Waterfall Methodology	60
3.1.2 Prototype	62
3.1.3 Activities In Each Phases of Waterfall Methodology	64
3.1.3.1 Analysis Phase	64
3.1.3.2 Design Phase	68
3.1.3.3 Implementation Phase	71
3.1.3.4 Testing Phase	72
3.2 Methods of Collecting Data	75
3.3 Requirements Analysis	78
3.3.1 Functional Requirements	78
3.3.2 Non-functional Requirements	80
3.4 Selection of Programming Language and Development Tools	82
3.5 System Requirements	83
3.5.1 Static and Dynamic Web Pages	83
3.5.1.1 Static Web Page	83
3.5.1.2 Dynamic Web Page	83
3.5.2 Creating Dynamic Web Page	84
3.5.2.1 Active Server Pages (ASP)	84
3.5.2.2 VBScript	84
3.5.2.3 Java Script	85
3.5.3 HTML	85
3.5.3.1 HTML 4.0	85
3.5.3.2 Cascading Style Sheet (CSS)	86
3.5.4 Client-Server Model	87
3.5.4.1 Server-side Scripting	90
3.5.4.2 Client-Side Scripting	90
3.5.5 Running ASP	91
3.5.5.1 Using Personal Web Server (PWS)	91
3.5.5.2 Using Internet Information Server (IIS)	92
3.5.6 World Wide Web (WWW)	92
3.5.7 File Transfer Protocol (FTP)	93
3.6 Summary	93

Chapter 4 System Design

4.1 Importance of System Design	94
4.2 Design Process	94
4.3 Functional Design	95
4.4 User Interface Design	96
4.4.1 Requester Interface	91
4.4.2 Administrator Interface	100
4.5 Database Design	104
4.5.1 Information Captured by The Database And Retrieved From Database	105
4.5.2 Tables	106
4.6 Program Design	111
4.6.1 EDDS Decomposition	111
4.6.2 EDDS Entity Relationship Diagram	112
4.6.3 EDDS Data Flow	114
4.6.4 EDDS Site Map	118
4.7 Expected Outcome	119
4.8 Summary	119

Chapter 5 System Implementation

5.1 System Implementation Process	120
5.2 System Coding	120
5.2.1 Methodology	120
5.2.2 Schedule	121
5.2.3 Hardware Requirements	121
5.2.4 Software and Development Tools Requirements	121
5.2.5 Unit Coding	122
5.2.5.1 ASP	123
i) Email	123
ii) Session Variables	125
iii) Redirect	126
5.2.5.2 HTML	126
5.2.6 Module Coding	129
5.2.7 Database Implementation	129
5.3 Summary	130

Chapter 7 System Evaluation

7.1 Importance of System Evaluation	147
7.2 Problems Encountered and Solutions	147
7.2.1 Development Tools Selection	147
7.2.2 Lack of Knowledge in ASP	148
7.2.3 Lack of SQL Basic	148
7.2.4 Lack of ASP Resources in University of Malaya	149
7.2.5 Lack of Hardware and Software Configuration Knowledge	149
7.3 Evaluation by End Users	149
7.4 System Strengths	150
7.4.1 User-friendly Interface	150
7.4.2 Easy to Use and Easy to Learn	150
7.4.3 Provide Database Access	151
7.4.4 Simpler Than Manual System	151
7.4.5 Less Error	151
7.4.6 System Transparency	151
7.5 System Constraints	152
7.5.1 Lack of Advance Security Features	152
7.5.2 Not a Full Working System, More to a Prototype	152
7.5.3 No Date and Time Line Validation	152
7.5.4 No Other Language Support	153
7.5.5 No Complex Error Detection	153
7.5.6 Email Attachment Using ASP	153
7.5.7 No Relationship Between Tables in Database	154
7.6 Future enhancements	154
7.6.1 Payment Module	154
7.6.2 More Categorized and Appropriate Presentation of Summaries	155
7.6.3 Source Email Table	155
7.6.4 Attachment/Upload Module	155
7.6.5 User Profile	156
7.6.6 Task List Layout and Presentation	156
7.6.7 Type of Document Detection	156
7.6.8 Other Language Support	156
7.6.9 Online Rules and Regulations for EDDS	157
7.7 Knowledge and Experienced Gained	157
7.7.1 Web Based Development	157
7.7.2 SQL Command	157
7.7.3 ASP	158
7.7.4 Development Tools	158
7.7.5 JavaScript	158
7.7.6 Developing a Medium Size Web Based System	158
7.7.7 Proper and Well Planned System Development	159
7.8 Summary	159
7.9 Conclusion	160

Appendices

A. User Manual	161
B. Administrator Manual	183
C. Questions Asked During System Testing	209
D. Coding	210
References	222

List of Figures

Figure 1.1	EDDS project schedule	8
Figure 2.1	University of Nevada, Reno Libraries screen shot	23
Figure 2.2	Byron State University Library home page screen shot	26
Figure 2.3	Stanford University Library home page screen shot	29
Figure 2.4	Arizona State University Libraries home page screen shot	32
Figure 2.5	Kent State University Library main page screen shot	35
Figure 2.6	Monash University Library main page screen shot	38
Figure 2.7	STP screen shot	41
Figure 2.8	MLSD screen shot	43
Figure 2.9	National Library of Medicine screen shot	45
Figure 2.10	The waterfall model	47
Figure 2.11	Arizona State University Libraries Document Delivery Services registration form	54
Figure 2.12	Nevada University Library request form	56
Figure 2.13	WUSC Document request form	57
Figure 3.1	The waterfall model	61
Figure 3.2	The waterfall model with prototyping	63
Figure 3.3	The server model and Peer-to-peer model architecture	87
Figure 4.1	Access to an article interface	98
Figure 4.2	Request for an article interface	98
Figure 4.3	Request for a chapter or book interface	99
Figure 4.4	Making enquiry interface	99
Figure 4.5	Giving suggestion interface	100
Figure 4.6	Layout interface	100
Figure 4.7	General book list interface	101
Figure 4.8	Generate report and summary interface	101
Figure 4.9	Giving feedback to user's enquiries interface	102
Figure 4.10	Giving feedback to user's suggestions and comments interface	103
Figure 4.11	Making query interface	103
Figure 4.12	EDDS decomposition diagram	111
Figure 4.13	EDDS Entity Relationship Diagram	113
Figure 4.14	DFD for EDDS scenario i: Requesting for materials	115
Figure 4.15	DFD for EDDS scenario ii: Materials arrival	116
Figure 4.16	OPD for EDDS scenario iii: Enquiries and suggestions made	117
Figure 4.17	EDDS site map	118
Figure 6.1	Logical Web Pages Flow for Registration Module	144

List of Figures

Figures	Description	Page
Figure 1.1	EDDS project schedule	8
Figure 2.1	University of Nevada, Reno Libraries screen shot	23
Figure 2.2	Syracuse University Library home page screen shot	26
Figure 2.3	Stanford University Library home page screen shot	29
Figure 2.4	Arizona State University Libraries home page screen shot	32
Figure 2.5	RMIT University Library main page screen shot	35
Figure 2.6	Monash University Library main page screen shot	38
Figure 2.7	SHPL main page screen shot	41
Figure 2.8	MUSC Document Delivery Services screen shot	43
Figure 2.9	National Library of Australia Loansome Doc screen shot	45
Figure 2.10	The Clement C. Maxwell Library screen shot	47
Figure 2.11	Arizona State University Libraries Document Delivery Services registration form	54
Figure 2.12	Nevada University Library request for a chapter of a book request form	56
Figure 2.13	MUSC Document Delivery categorize forms web page	57
Figure 3.1	The waterfall model	61
Figure 3.2	The waterfall model with prototyping	63
Figure 3.3	Client-server model and Peer-to-peer model architecture	87
Figure 3.4	Requesting for a dynamic web page	88
Figure 3.5	Server-Side Scripting processes	89
Figure 4.1	Login interface	97
Figure 4.2	Menu interface	98
Figure 4.3	Request for an article of a journal interface	98
Figure 4.4	Request for a chapter of a book interface	99
Figure 4.5	Making enquiries interface	99
Figure 4.6	Giving suggestions interface	100
Figure 4.7	Logout interface	100
Figure 4.8	Generate task list interface	101
Figure 4.9	Generate report and summary interface	101
Figure 4.10	Giving feedback to user's enquiries interface	102
Figure 4.11	Giving feedback to user's suggestions and comments interface	103
Figure 4.12	Making query interface	103
Figure 4.13	EDDS decomposition diagram	111
Figure 4.14	EDDS Entity Relationship Diagram	113
Figure 4.15	DFD for EDDS scenario i: Requesting for materials	115
Figure 4.16	DFD for EDDS scenario ii: Materials arrival.	116
Figure 4.17	DFD for EDDS scenario iii: Enquiries and suggestions made	117
Figure 4.18	EDDS site map	118
Figure 6.1	Logical Web Pages Flow for Registration Module	144

List of Tables

Tables	Description	Page
Table 2.1	Web page features summary table	52
Table 3.1	Summary of analysis phase	65
Table 3.2	Summary of design phase	69
Table 3.3	Summary of implementation phase	71
Table 3.4	Summary of testing phase	72
Table 3.5	Types of testing and the corresponding action and responsible team for each action	74
Table 3.6	Differences between Client-Side Scripting and Server-Side Scripting	90
Table 4.1	User requirements	95
Table 4.2	Administrator requirements	96
Table 4.3	Different functions or sub-functions of EDDS	105
Table 4.4	User login function data types and attributes	106
Table 4.5	Request for an article o a journal function data types and attributes	106
Table 4.6	Request for a chapter of a book function data types and attributes	107
Table 4.7	User enquiry function data types and attributes	107
Table 4.8	User suggestion and comment function data types and attributes	108
Table 4.9	Task list generating function data types and attributes	108
Table 4.10	Summary generating function data types and attributes	109
Table 4.11	Enquiries feedback function data types and attributes	109
Table 4.12	Suggestions and comments feedback function data types and attributes	110
Table 4.13	Information query function data types and attributes	110

1.1 Introduction

Every year there has been an increase in the enrolment of students at the University of Malaya and the rise in escalating demand for information. One of the main sources of information for our campus besides the Internet is the University of Malaya Library. The library has kept serving the students of the campus without any disappointment for the past few decades. As for the explosion of the computer and information technology is forcing a change in the way information should be delivered and available to users.

Chapter 1

Many are aware that postgraduates program is available at the University of Malaya. Mostly the programs are based on dissertation and this means that the student task and assignment will concentrate more on doing research and developing new concepts and systems. In order for them to perform research and development work, they need information, which they can find from various sources.

Introduction

It is the responsibility of the library to create a new environment in the campus. Some of the steps taken are automating the existing information systems, developing new systems and integrating new systems with the old ones.

The concept of electronic library and digital library is no more a fantasy in overseas educational organizations. Trends are evolving towards this future environment. One of the steps in bringing our campus closer to this environment is having our own electronic document delivery services (EDDS). From the research conducted, EDDS have been implemented in overseas institution and they are evolving from time to time to cater the ever-increasing demand of information request. Although there are several undeniable non-local working systems available, but their features are not all suited to our environment, need and expertise. This project will focus on developing an EDDS for the University of Malaya Library.

1.1 Introduction

Every year there has been an increase in the enrolment of students at the University of Malaya and this lead to escalating demand for information. One of the main sources of information in our campus besides the Internet is the University of Malaya Library. The library has been serving the citizens of the campus without any disappointment for the past few decades; however the explosion of the computer and information technology is forcing a paradigm shift in the way information should be delivered and available to users.

Many are aware that postgraduates degree program is available at the University of Malaya. Mainly the programs are based on dissertation and this means that the student task and assignment will concentrate more on doing research and developing new concepts and systems. In order for them to perform research and development work, they need information, which they can find from various sources.

In parallel with this new demand, the Faculty of Computer Science and Information Technology, University of Malaya, having the infrastructure of the new era of technology is responsible to create a new environment in the campus. Some of the steps taken are automating the existing information systems, developing new systems and integrating new systems with the old ones.

The concept of electronic library and digital library is no more a fantasy in overseas educational organization. Trends are evolving towards this future environment. One of the steps in bringing our campus closer to this environment is having our own electronic document delivery services (EDDS). From the research conducted, EDDS have been implemented in overseas institution and they are evolving from time to time to cater the ever-increasing demand of information request. Although there are several undeniable non-local working systems available, but their features are not all suited to our environment, need and expertise. This project will focus on developing an EDDS for the University of Malaya Library.

1.2 What is Electronic Document Delivery Systems?

EDDS is a web based system that promises user a quick and efficient way in accessing information. Users no longer need to go to the library to make a request for materials. Instead, users can make their request from home or even work place. This is possible with the use of computer technology to facilitate and automate the procedure. Users need to access the system and log in their request and wait for the librarian to process the request, in the end the document will be sent directly to the user electronically.

There are several definitions about what an EDDS is. Some of the definitions are as follow:

1. Electronic document delivery is “... *the use of electronic technologies in support of the interlending activities of libraries.*” [1]
2. Electronic document delivery is “... *the transmission of documents on request from network databases.*” [2]

1.3 The aim

The aim of this project is to automate the document delivery process in order to provide more efficient services. Automating the existing process will speed up user's information request. By using the computer technology, the processing of user's information request can be carried out more efficient and in order. The use of database can help librarian to retrieve user's information, without needing to browse through a pile of forms in order to get particular information. This system is predicted to make the document requesting services available 24 hours per day and 7 days per week because the system will be available online. Because the system is available online, a user can actually reach the system using a remote computer without going to the library him or herself just to use the system. This will be a relief to users because it is very hard to find parking space available in the library parking lot. Further more, the system is more time saving and user can have extra time to do their own work. On the other hand, the librarian need not waste

their time in handling the submission of forms or getting the form for the users. The librarian only need to spend their time in processing and servicing the request only by looking at task list generated by the database at end of the day. This system is also is the first step in moving the library towards a paperless environment and creating a digital library that will offer more services and value for money to their customers.

1.4 Project Objectives

Below are the objectives of EDDS project:

- To create an automated document delivery system for the University of Malaya Library
- To create a user interface to help user to navigate through the requesting stage in order to get their document(s).
- To create an e-form to replace the current hardcopy form which will be used to capture the user's information and request
- To develop a database to store the user's information captured through the e-form
- To create interface for a request for an article from a journal
- To create interface for a request for a chapter of a book

1.5 Project Scopes

The implementation of this project would be limited to automating the requesting process in the document delivery services. This is because, currently the University of Malaya Library does not have any services in delivering document requested via electronic form. The articles are mainly in the form of hardcopy and the requester or user need to go to the library to collect their document. This will be a future consideration that the documents delivered to the users are in electronic form such as in PDF format. And the document or the link to the document will be directly sent to the user by e-mail. In such way the whole process will gain its benefit in providing more efficient, time saving and cost effective services.

The types of document that can be requested are articles from a journal and chapter(s) of a book. These are the type of document mainly requested by the users.

The documents that are to be requested must be first be proved that the article is not found in the University of Malaya library. Further search must be carried out to identify the information of the articles such as the title, author and year the article is published before the user can actually use this system to request for the article. The library's librarian will do this search. Besides than that, users themselves can search the library online catalog (OPAC) to make sure that the material is not available in the library collections.

The system is available to help the users in their research and find outs, so the library restricted this service only to post graduate students and academic staff.

1.6 Project Significances

Below are the significances of the project.

1.6.1 To ease the effort to obtain information

The effort needed to request and obtain the material, which is not found in the library collections, will be reduced. Users no longer need to go to the library to place a request.

1.6.2 To promote more research work to be done

Indirectly, this will promote more research work to be done because it is easier to request for information using the EDDS compared to the current manual system.

1.6.3 To help the library of UM to provide more efficient services

The librarian only processes the requests at the end of the day or the next day morning. Librarian can concentrate on other tasks and services during the daytime rather than entertaining user's request(s).

1.6.4 To help the library's officer to plan for future services for document delivery services

The EDDS could generate reports and summary of the service usage; receive enquiries and suggestions from users. This will help the library to plan for new services and schedule staff to process request during certain time zone where request can be tremendous.

1.6.5 To save the amount of time needed in making request and to process the requests

The amount of time needed for users to place request will be reduced. At the same time the time needed by librarian to process the request will also decrease. Librarian no longer need to retype or re-enter user's information and request. The time needed for the materials to be sent to users will also decrease by using the e-mail.

1.7 Hardware and software requirement

The hardware and software that is needed to develop this system are listed below.

Pentium III or compatible PC based computers, 500 MHz processors, 256 MB Ram, display card, sound card.

Windows 2000 Server (Server side)

Windows 2000 (Client Side)

Authoring tools

Programming languages

Authoring tools and programming languages will be used to develop the user interface; design the web page and also develop dynamic content for web based solution.

Each step will be conducted in different phase and path. Some of the phases can be overlapped and some of them cannot. Figure 1 presents the project schedule of EDDS.

1.8 Project schedule

In developing a system, the following series of activities will be carried out.

1. Design Phase
 - i. Requirements analysis
2. Analysis Phase
 - i. System design
 - ii. Program design
3. Implementation
 - i. Coding
4. Testing
 - i. Unit and integration testing
 - ii. Acceptance testing
5. Delivery
 - i. Operation and Maintenance
6. Documenting
 - i. Documentation

Each step will be conducted in different phase and path. Some of the phases can be overlapped and some of them cannot. Figure 1 presents the project schedule of EDDS.

Project Schedule

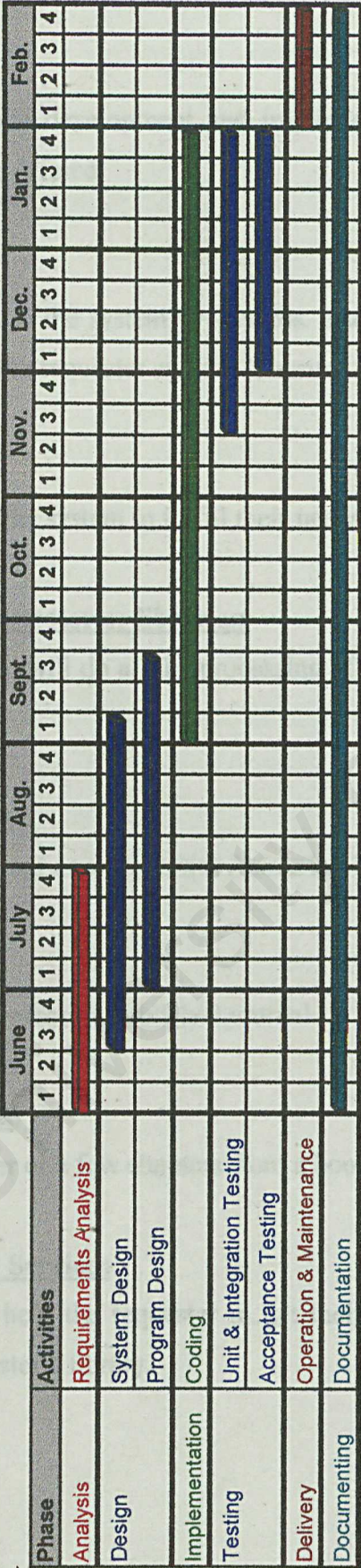


Figure 1.1 EDDS project schedule

1.9 Definition

Throughout the whole development and implementation of the system there are few terms that need to be defined.

User

A person who will use the system or services. The users for the system will be divided into two categories, the requester and the librarian.

Requester

Person who will use the system to fulfill their needs.

Personnel or administrator or librarian

Person or people who will do all the processing of requester request and maintaining the system.

Document or material

Can be article of journal or a chapter(s) from a book.

Article

Referring to an article from a published journal.

Chapter(s) of a book

Referring to a chapter or a few chapters from a book and not the whole book itself.

Document Delivery Services

A service that will help the requester to get their information, which is not found, or possess by the requester's library.

Electronic Document Delivery System

A system that automates the manual document delivery services using information technology infrastructure.

Electronic Document Delivery Services

Document delivery services using information technology and computer technology such as email and Internet.

Inter Library Loan

Service that enable materials to be loaned from other libraries. Materials will be sent to the user and need to be returned.

1.10 Summary

In this chapter, the objectives, scopes, aims and limitations; and the significances of the EDDS project have been highlighted. Besides than that, the hardware and software requirements to develop the system are also listed. The whole project started early of June 2001, and will follow a series of development activities in the development process. Series of activities are being matched with their corresponding development stage of software development cycle. Lastly, the terms, which will be encounter throughout the whole development project, are defined.

Chapter 2

Literature Review

2.1 Importance of Literature Review

Literature review is an important process in system development. There are a lot of activities in this stage such as facts finding, summarizing, analysis and synthesis of the features and modules that the future system built will have. This is to ensure a better understanding of the system that will be developed and to choose the best way to plan and organize work to achieve the objectives of the system.

Researches have been carried out to find and try out existing and relevant system that is available elsewhere. This is because it is hard to find any system that is being used in Malaysia, so choices of existing system are available at overseas institution. This includes research on the information about the system that are available on the web page. Besides than that, research also conducted to examine available online forms. Furthermore, reports and articles about the available system also being study and this information have helped in creating a more accurate and usable system.

Besides looking at the information available and the example of forms, a review also being done to comment on the look, layout and approach being taken in organizing the information and location of the information. The features available on each website and relevant links are also being evaluated. By doing evaluation, the system that will be developed, will inherit all the style that is more pleasant to users. This also to motivate users in using the future developed system by implementing guides and links to help pages.

2.2 Document Delivery Services Review

Document delivery services is used by libraries to help their user in getting their needed information which can not be found in their respective library. Source of information could be the nearby libraries or institution spanning to abroad institutions. The service usually is a manual system where users need to go to the library to make request using the request form.

Upon receiving the request form by the librarian from the user, the librarian will have to make sure that the information needed is not available in the library before proceeding with searching the nearest source. This is crucial because to obtain information from outside when the information is available internally is a waste of time and human resources. The librarian work does not end here, instead he or she need to search for the nearest source to save delivery time and delivery cost. When the source has been identify, the librarian need to make a request and payment if it is involved.

The institution that has the source will send the information by snail mail or fax to the requesting library after settlement of way of payment made. When the information arrived at their destiny, the librarian will have to notify the requester by mail, fax, phone or e-mail. The requester then has to go to the library to collect the information and is subject to payment, library rules and regulations.

The material or information can be kept by requester or sometimes need to be return back to the source. The whole process itself is manual and time consuming. This system eventually emerged itself to be on the same track with the advancement of information technology.

2.3 Electronic Document Delivery Services Review

The electronic document delivery services in the long run have evolved from the tradition document delivery services. Most of the available services evolved from the tradition services. The concept is the same but the only difference is the system is fully automated and digitizes using information technology and computer technology. All the users need to do is fill up a web based request form and wait for the information to arrive at their computer desktop or workstation.

The librarian will have to view the request list and locate the nearest source and forward that information to the source institution. Responding to the request, the source institution will send back the needed information to the requester instead of the requesting library. Some of the approach taken is requester will be notify by electronic message upon the availability of the information to be viewed. This is applicable on information or materials, which are non-returnable. Some information source like Medical University of South Carolina (MUSC) have way to allow limited number of viewing or limited duration where users are allowed to posses the information. The system itself will then removed from the system.

2.4 Benefits of using Electronic Document Delivery Services (EDDS).

Based on personal evaluation, the following have been seen as the benefits of EDDS.

1. The time needed to process and deliver the information to the requester will be shortened when using the EDDS.
2. The number of resources needed to process a request is lower when using the EDDS.
3. EDDS promotes a paperless environment.
4. Request can be made anytime anywhere because the system is available daily around the clock.
5. The cost of transporting or sending the information to the requester will be minimized.
6. EDDS encourages researchers to gain more information.
7. EDDS is a new era of DDS where it can provide services provided by DDS and yet offer a more value added services.

2.5 Evolution of electronic document delivery systems

By considering the evolution of document delivery systems from the start, it is possible to identify several generations: from mainly manual to highly electronic. Every generation is based on a more refined approach due to improvements in technology.

In the present library world three generations can be distinguished: systems based on online ordering, non-integrated supply driven image-based systems and integrated stand-alone image-based systems. [3]

2.5.1 First generation: Online ordering

If easiness and speed in document delivery is the issue, the natural starting point for any document delivery services are reference databases. They contain all the bibliographic information needed to generate a request in an unambiguous way. Users need to complete the request by adding his personal data, which to a large extent may already be present in administrative databases. An application connected to the reference databases could produce a work list for document delivery personnel, which makes photocopies. Final delivery is established by sending the documents by ordinary (snail) mail or facsimile to the applicants.

Currently, such a system is implemented in the Excerpta Informatica databases in Tilburg while all major host organizations, like DIALOG and ESA/IRS, offer these kinds of services.

An important drawback in this generation is the lack of efficiency in request processing. Each time an article is requested library personnel has to go to the shelves, take the journal issue or bound volume, and make photocopies, even if it is a request for an article demanded before. The process is very labor intensive, but easy to implement and standards are hardly called for. [3]

2.5.2 Second generation: Supply driven image based systems

The second generation document delivery systems resolves the efficiency problem to a certain degree by introducing a supply driven approach and by storing articles as images (in bitmap or other format). The advancement of scanners technology has make scanner become more popular and act almost similar to photocopying, the most efficient way is to bitmap the articles pages. In this second generation, standards for storage and retrieval of image files become of crucial importance.

This approach advocates the scanning and storage of articles, which are expected to be demand frequently, in advance. An application can be easily build in such a way as to automatically produce a hard copy together with a header page containing the address of the applicant which can again be send by snail mail or facsimile. If there is a request for articles, which have not been stored in advance, it will be process using the first generation systems.

The ADONIS system resembles to a large extent a second-generation document delivery system. [3]

An important property of third generation systems is scalability. Due to the demand driven approach, one only need to scan articles, which are actually in demand. In this respect, a maximum flexibility in human resources as well as equipment is accomplished. An important drawback of an image-based system is that it is not open to likely future developments. A desirable development is that more and more printed material will become available in machine-readable form. In the publisher's world, for instance, SGML is slowly but persistently making its way as a standard, while in the office world ODA is becoming increasingly important. From the viewpoint of libraries, with their close relation to publishers, SGML is probably the standard to be reckoned with in the near future. Applications for electronic document delivery should therefore be able to deal with different document representations.

2.5.3 Third generation: scalable stand alone systems

Third generation systems can be characterized as scalable stand-alone systems. These systems are still based on images, but a demand driven approach is chosen to reduce overhead, i.e. only those articles in demand will be scanned. Furthermore, the delivery service is expanded with delivery at the end users workstation. Integration of the system within the library environment is enhanced by the incorporation of several local reference databases. However, the system still has a stand-alone character.

An example of an approach having second and third generation characteristics is the Mercury project of Carnegie Mellon.

A supply driven approach is combined with tight integration with the reference databases. If a user finds a reference, the system shows if images of the referenced article are available. These images can be presented on the workstation by clicking a button. An example of a demand driven approach is the NAL/NCSU digitized document transmission project, although this system seems not very well integrated with other library systems.

An important property of third generation systems is scalability. Due to the demand driven approach, one only needs to scan articles, which are actually in demand. In this respect, a maximum flexibility in human resources as well as equipment is accomplished. An important drawback of an image-based system is that it is not open to likely future developments. A plausible development is that more and more printed material will become available in machine-readable form. In the publisher's world, for instance, SGML is slowly but persistently making its way as a standard, while in the office world ODA is becoming increasingly important. From the viewpoint of libraries, with their close relation to publishers, SGML is probably the standard to be reckoned with in the near future. Applications for electronic document delivery should therefore be able to deal with different document representations.

A second drawback is noticed in the non-integrated character of Inter Library Loan services and existing document delivery procedures. Functionality could be enhanced for both users and the library. For users integration means that via one channel they are able to make both internal and external (ILL) requests for articles. Integration for the library means that all document delivery activities are passing one department. Both internal and external (ILL) requests should be received at one point and processed by one organizational unit.

The factors mentioned here call again for standards, first of all standards for the generation of requests and secondly a standard for unique article identification. [3]

2.6 Electronic Document Delivery System in Malaysia

Until today from my own facts finding using the Internet, it is very hard to find an existing system being used in Malaysia. So the available systems used for literature review are from the universities abroad. And yet the libraries abroad are using DDS and it is not a true EDDS. It is seen that the DDS is actually a starting point to implement EDDS. For example, RMIT University will be introducing their own EDDS services in complementing their DDS services.

2.7 Current Available Services

Mainly the libraries use the document delivery service. A research has been conducted to find more information about the existing system. The main sites to search for existing features are the University Libraries around the world. The searching came across six Universities Library, which offer the service. Please take notes that most of the institution reviewed provides DDS. The institutions mentioned are:

1. University of Nevada Library
2. Syracuse University Library
3. Stanford University Library
4. Arizona State University Libraries
5. RMIT University Library
6. Monash University Library
7. SHPL
8. Medical University of South Carolina (MUSC) Library
9. National Library of Australia
10. The Clement C. Maxwell Library

The above-mentioned libraries, explain the services using their own respective library web page. The information of services is available under the services part.

Thesis

This option is for user from other institution who wishes to obtain thesis from the library. Or either it can be thesis from other libraries, which the institution user would like to obtain.

How to place a request

Way to make a request and to whom the request is made.

2.8 Types of information available on web pages about

Document Delivery Services

The types of information that can be viewed are as follow. Each of the library have own unique information, but not the least, most of the information display are typical such as.

Information about the services

The particular web page that explains about the services begins with a short explanation on what a document delivery services is. The main point of the service is about obtaining needed materials not available at the respective institution.

Eligibility for the services

Highlighting the group of people who is eligible to use the services.

Type of materials can be requested

Explain about the type of materials that can be requested through the services.

Type of materials which cannot be requested

Explain about the type of materials, which cannot be requested.

Thesis

This option is for user from other institution who wishes to obtain thesis from the library. Or either it can be thesis from other libraries, which the institution user would like to obtain

How to place a request

Way to make a request and to whom the request is made.

Source of materials

Explain where the materials will be sourced

Number of requests

This section explains about the maximum number of request that can be made by a user according to rules fixed by the library administrator.

How long it takes to receive materials

The period of delivery is the time needed for the materials to arrive after a request is submitted.

Costs

This part explains about the charges, which users need to pay when using the services.

Payment

The payment terms are explained in this section.

Notification when materials arrive.

Way how the user will be contacted as soon as the materials are available for collection.

Delivery options

This section explains about the delivery method that a user can choose according to their preference.

Checkout period

The duration of which a material can be obtain by the user.

Overdue Materials

Explain the assessment when a material obtains by user is over than the allowable period.

Lost or Damaged Materials

Terms and condition that user need to obey if materials are found lost or damage during user's possession.

Article not picked up

Subject to any fees accrued for materials whether they are pick up or not.

Library user's responsibilities

This section highlights the user's responsibilities.

Library Responsibilities

This section explains the responsibilities of the library document delivery service.

Further enquiries

List of contacts where further information can be acquired if any doubts arise.

Alternative option to obtain materials not in print

Explain how user can obtain materials not in print from digital source.

Copyright Regulations

Explain about the copyright act for their respective country, which the institution resides.

Electronic Document Delivery System
Chapter 2 – Literature Review

1. University of Nevada Libraries

URL : www.library.unr.edu/depts./docdel/default.htm

Date accessed : 20 July 2001



Document Delivery Services

Document Delivery Services aids University of Nevada, Reno faculty, staff, and students in their study and research by obtaining needed materials not available at the University from other libraries and commercial document suppliers.

- [Services](#)
- [Who is eligible?](#)
- [FAQ--Frequently Asked Questions About Our Services](#)
- [Interlibrary Borrowing Policy](#)
- [How to submit requests for materials on Interlibrary Loan](#)
- [Go To Request Forms](#)

- [Information for Other Libraries, Including Lending Policies](#)

For further information contact:

Document Delivery Services
Getchell Library/322
University of Nevada, Reno
Reno, Nevada 89557
PHONE: (775) 784-6500, ext. 326
FAX: (775) 784-1751
E-MAIL: ill@admin.unr.edu

Hours: 8:00 a.m. - 5 p.m., Monday - Friday

Please direct questions to:
[Millie Syring](#)

The University Libraries/322
1664 N. Virginia St.

Figure 2.1 University of Nevada, Reno Libraries screen shot [4]

The University of Nevada Libraries Library web page contains information about:

1. Information about the services
2. Eligibility for the services
3. Type of materials can be requested
4. Type of materials which cannot be requested
5. How to place a request
6. Number of requests
7. How long it takes to receive materials
8. Costs
9. Payment
10. Notification when materials arrive.
11. Delivery options
12. Checkout period
13. Overdue Materials
14. Lost or Damaged Materials
15. Article not picked up
16. Library user's responsibilities
17. Library Responsibilities

Opinion about the website

The University of Nevada Libraries Library Web page provides some functions such as links to

- | | |
|-----------------|-----------------|
| i. Home | vii. Subjects |
| ii. Search | viii. Databases |
| iii. Site index | ix. eJournals |
| iv. Requests | x. eBooks |
| v. Help | xi. Reference |
| vi. Catalog | |

Figure 2.1 shows the University of Nevada Libraries web page screen shot. The links are groped into a bar and it is located at the top of the web page. This approach is good because the user can view the links at the top of the web pages, as they will tend to start their reading from top to bottom. It is also a great effort to provide all those links that will eventually help users to get all the resources and help they need. The services web page organizes information about the type of services and related information in bulleted list that will make it more in order and neat. This will help the user to find relevant information easily. The combination of colors shows a bit of contrast by using yellow to highlight some of the links first alphabet. This will make the user more alert to what type of links provided.

The website offers a systematic way to look for the type of information needed. The combination of colors also soothes the eyes. Further explanations about the services are located in separated web page. This approach will make the main page more compact and brief without congested it with too much information. Users have the option to view the information that they want rather than browsing every single piece of information available on the whole library web page. There is also a link to the request form which user can used to send request for materials.

Figure 2.2 Syracuse University Library home page screen shot [5]


2. Syracuse University Library

URL : <http://libwww.syr.edu/>


Date accessed : 20 July 2001

[Catalog/Databases](#)
[General Information](#)
[News/Events](#)
[Services/Policies](#)
[Instruction/Training](#)
[Digital Projects](#)

[SUMMIT](#)
[Databases](#)
[Request Forms](#)
[Hours of Service](#)
[HOME](#)

 **SYRACUSE UNIVERSITY LIBRARY**

Services & Policies

- [Service Request Forms](#)
- [Circulation](#)
- [Copy Services](#)
- [Dissertation/Thesis Binding](#)
- [Interlibrary Loan/Document Delivery](#)
 -  [Login to ILLiad](#)
- [Media Services](#)
- [Reference Services](#)
- [Reserves](#)
- [Independent Study Degree Program \(ISDP\)](#)

[Catalog/Databases](#) || [General Information](#) || [News/Events](#) || [Instruction/Training](#) || [Digital Projects](#)
[SUMMIT](#) || [Databases Main Menu](#) || [Request Forms](#) || [Hours of Service](#) || [Syracuse University](#)

© 1998 - Syracuse University Library
Send comments to the [Suggestion Box](#)
URL: <http://libwww.syr.edu/services/index.html>

Figure 2.2 Syracuse University Library home page screen shot [5]

The Syracuse University Library web page contains information about:

1. Information about the services
2. Eligibility for the services
3. Type of materials can be requested
4. Source of materials
5. How to place a request
6. Number of requests
7. How long it takes to receive materials
8. Costs
9. Notification when materials arrive.
10. Delivery options
11. Checkout period
12. Overdue Materials
13. Lost or Damaged Materials
14. Article not picked up
15. Further enquiries
16. Alternative option to obtain materials not in print

Opinion about the website

Figure 2.2 shows the Syracuse University Library home page screen shot. The library website organize its links to other relevant pages with different approach. The types of information links available are:

- i. Catalog/Databases
- ii. General Information
- iii. News/Events
- iv. Instruction/Training
- v. Digital Projects
- vi. SUMMIT
- vii. Databases Main Menu
- viii. Request Forms
- ix. Hours of Service
- x. Syracuse University

The links are organizes in a list located at the lower part of the websites. It also has the same links located in a smaller window at the right hand side. The smaller window has a different background color to attract users attention and to separate the information part and links. The combinations of colors used are contrasting and the fonts used and size used is more to formal presentation.

The services and policies page have all the services provided linked to it. Users do not need to go to another page to get other services but they can still access other services using the same page. The user login link is also available under the interlibrary loan/document delivery link. User can straight away proceed to the login screen if they do not want to view the information about interlibrary loan/document delivery information. The web page presents its content in a very straight forward and organizes way. By this way, users can examine the page more easily.

Figure 2.3 Stanford University Library home page screen shot (6)

3. Stanford University Library

URL : <http://www-sul.Stanford.edu>

Date accessed : 20 July 2001


Catalog

Search

Home

Tell Us

Help



User Services

[Circulation](#) || [Course Reserves](#) || [Reference and Instruction](#) || [Interlibrary Services](#) || [Facilities](#)

Circulation and Access

- ▶ [My Circulation Records](#) (view current checked out items, fines, etc.)
- ▶ [Circulation and Access](#)
 - ▶ [Access Policies](#)
 - ▶ [Circulation Policies](#)
 - ▶ [Policy on Proxies](#)
 - ▶ [Use of Libraries by Institutions](#)
 - ▶ [Fees for Purchased Cards](#)
- ▶ [Services for Disabled Users](#)
- ▶ [Stanford Card Plan](#) (DART machines)

Course Reserves

- ▶ [General Information](#)
- ▶ [Guidelines for Submitting Reserve Materials: \[Media\]\(#\) & \[Print\]\(#\)](#)
- ▶ [Reserves Catalog](#)

Facilities and Equipment

- ▶ [Library Hours, Maps, and Directories](#)
- ▶ [Physical Access for Disabled Users](#)
- ▶ [Computer Clusters](#)
- ▶ [Late Night Study Areas](#)
- ▶ [Photocopy Services](#)
- ▶ [Network Connections for Your Laptop in the Libraries](#)

Interlibrary Services

- ▶ [Interlibrary Services](#)
 - ▶ [General Information](#)
 - ▶ [Interlibrary Borrowing](#)
 - ▶ [Cooperative Agreements](#)
 - ▶ [Document Delivery Online Request Forms](#)
 - ▶ [Interlibrary Lending Policy](#) (for non-Stanford users)

Reference and Instruction

- ▶ [Beginning Library Research](#)
- ▶ [General Reference Resources](#)
- ▶ [Instruction](#)
- ▶ [Scholars' Workshops](#)
- ▶ [Ask a Librarian](#)

Figure 2.3 Stanford University Library home page screen shot [6]

The Stanford University Library web page contains information about:

1. Information about the services
2. Eligibility for the services
3. Type of materials can be requested
4. Type of materials which cannot be requested
5. How to place a request
6. Notification when materials arrive.
7. Delivery options

Figure 2.3 shows Stanford University Library web page screen shot. The web site is well organized with links to different information. For example, if a user would like to use the services, they only need to click the services link rather than to have all the information needed dumped into a website with scroll bar to view. The web pages contain only the information that it should have, rather than irrelevant information. The combination of different colors to indicate different services provide simple and attractive layout because the colors are not overdone and pleasant to look at. It also organizes the layout in a well-manage form.

Opinion about the website

1. The web page provides links to Catalog, Search, Home, Tell Us and Help pages arranged in a bar located on the bottom of the page.
2. Besides than those links mentioned above, there is also a minor link bar, providing link to other pages in the library homepage such as Circulation, Course Reserves, Reference and Instructions and Interlibrary Services Facilities.
3. On the Services page itself, it also contains a lot of links which is categorizes into type of services in a list form.
4. The combination of colors used for the web page is simple and it reflects its purpose of delivering information about services to viewer.
5. The type of fonts and fonts size used is more formal and neat. This will make reading easier and looking for the needed information easier.
6. The web pages for services is grouped into a individual page so that user can view the information about the services when they really need to and not forcing the user to view if the information is irrelevant or not in much use for the user.

Figure 2.3 shows Stanford University Library web page screen shot. The web site is well organized with links to different information. For example, if a user would like to use the services, they only need to click the services link rather than to have all the information needed dumped into a website with scroll bar to view. The web pages contain only the information that it should have rather than irrelevant information. The combination of different colors to indicate different services provide simple and attractive layout because the colors are not striking and pleasant to look at. It also organizes the layout in a well manage form.

Figure 2.4 Arizona State University Libraries homepage screenshot [7]

4. Arizona State University Libraries

URL : <http://www.asu.edu/lib>

Date accessed : 20 July 2001

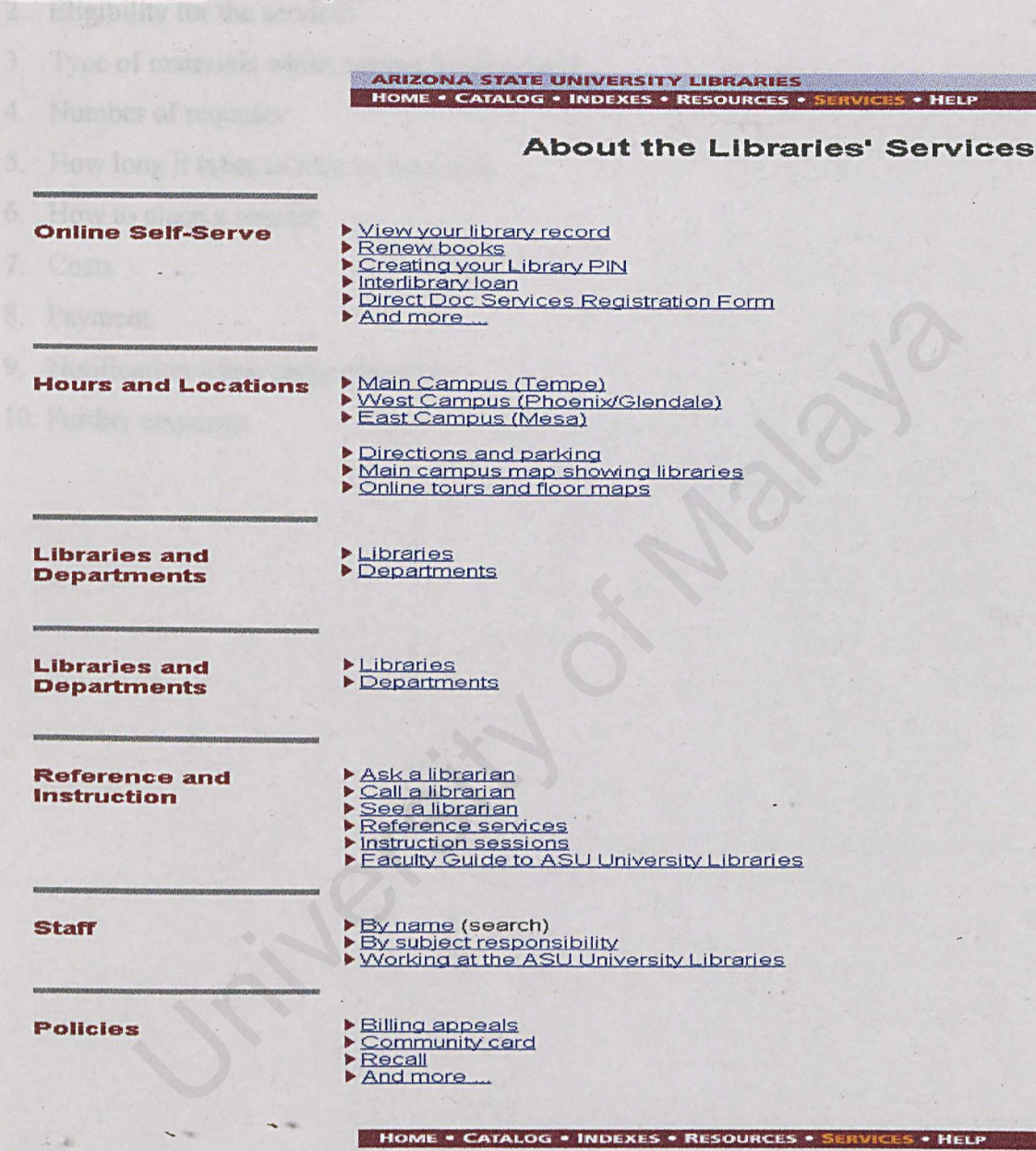


Figure 2.4 Arizona State University Libraries home page screen shot [7]

Arizona State University Libraries web page contains information about:

1. Information about the services
2. Eligibility for the services
3. Type of materials which cannot be requested
4. Number of requests
5. How long it takes to receive materials
6. How to place a request
7. Costs
8. Payment
9. Notification when materials arrive
10. Further enquiries

Opinion about the website

The Arizona State University Libraries web page has the links bar located at the top and bottom of the page. The links are

- i. Home
- ii. Catalog
- iii. Indexes
- iv. Resources
- v. Services
- vi. Help

Figure 2.4 shows Arizona State University Libraries web page screen shot. Because the page is long, so the bar bottom bar will eventually save the users effort to scroll upwards to reach the links. The can use the links at the bottom of the page as it is near and visible. The web page uses simple combination of colors and formal fonts. Red is used to highlight the services followed by a line to separate different services so that the types of same services are grouped together. This will help user in recognizing what are the services provided. The Interlibrary loan and the online Direct Doc Services Registration Form are separated into two different links. This would save the amount of time users need to spend in viewing the information about the services if they are already familiar with the system.

5. RMIT University Library

URL : <http://www.lib.rmit.edu.au/dds>

Date accessed : 20 July 2001

RMIT University Library

newsnewsnews
GREAT NEWS about REMOTE ACCESS!

Catalogue access... RMIT University Library and other library catalogues

Information resources... journal article databases, e-journals, general reference

Library services... borrowing, Reserve services, Document Delivery and more...

Learning Centres... information about our computer facilities

Help using the Library... online tutorials, subject guides and more...

Forms and requests... shortcut to a selection of online forms

About the library... library websites, contact Liaison Librarians and more...

- locations and opening hours
- website map

HELP US IMPROVE OUR WEBSITE!

INFORMATION LITERACY
RMIT UNIVERSITY

?

new DIGITAL resources

comments and suggestions

© 1999-2001 RMIT University PO Box 2476V Melbourne 3001 AUSTRALIA
RMIT Library site is maintained by: manager@www.lib.rmit.edu.au
Last updated: Wednesday, 1 June 2001
[disclaimer]

RMIT UNIVERSITY

Figure 2.5 RMIT University Library main page screen shot [8]

Opinion about the website

Figure 2.5 shows the RMIT University Library web page screen shot.

RMIT library main page has a very sophisticated look and feel due to the combination of

The RMIT University Library web page contains information about:

1. Information about the services
2. Eligibility for the services
3. Costs
4. How to place a request
5. How long it takes to receive materials
6. Delivery options
7. Number of requests
8. Thesis
9. Enquiries
10. Comments and suggestions

- i. Information resources
- ii. Library services
- iii. Learning center
- iv. Help using the library
- v. Forms and requests
- vi. About the library

- vii. Location and opening hours
- viii. Website map
- ix. Information literacy
- x. e-query
- xi. New digital resources
- xii. Comment and suggestions

Opinion about the website

RMIT library main page has a very sophisticated look and feel due to the combination of color of its background and the fore ground color. The colors are contrasting and it does help in making the information delivery clear. The combination shows a modern and high tech look, appropriate with its digital services theme. It has some multimedia element in it. The whole page requires much longer time to download compare to the previous web pages. This is not a good approach because it is very frustrating to download this page during busy network traffic. It also has a site map to help users to navigate through their web pages. The fonts type and size used is formal except for some element, which has a stylish design.

The RMIT University Library web page has the links, which are:

- | | |
|---------------------------|-----------------------------------|
| i. Catalog access | viii. Locations and opening hours |
| ii. Information resources | ix. Website map |
| iii. Library services | x. Information literacy |
| iv. Learning center | xi. e-query |
| v. Help using the library | xii. New digital resources |
| vi. Forms and requests | xiii. Comment and suggestions |
| vii. About the library | |

6. Monash University Library

URL : <http://www.lib.monash.edu.au/docdel>

Date accessed : 20 July 2001

MONASH HOME > MONASH INFO > NEWS & EVENTS > CAMPUSES & FACULTIES > MONASH UNIVERSITY LIBRARY

Catalogue > Databases > Subjects > Services > Collections > Virtual Librarian > About the Library >

MONASH UNIVERSITY

Services

Monash University Library offers its staff and students a huge range of services. Students of other higher education institutions may access some collections through reciprocal borrowing schemes such as [CAVAL](#), please enquire at your home library first. [MONINFO](#) provides library access for corporations, organisations and professionals who need library services.

Borrowing

- [Borrowing from Monash libraries](#)
- Borrowing from non-Monash libraries
 - in person via [CAVAL](#)
 - [InterLibrary Loans](#) (via DocDel)
 - other [Australian libraries](#)
- [Borrowing rights for visitors](#)

Using the library

- [Photocopying](#)
- [Computer facilities](#)
- [Study areas](#)

Remote library services

- Frequently asked questions
- [Connecting to electronic databases and journals](#)
- [Electronic requests](#)
- [Flexible Library Services Unit](#)

Inquiries & assistance

- Email inquiry service ([e-Query](#))
- Telephone inquiry service ([MULTELS](#))
- Online tutorials ([Virtual Librarian](#))
- Printed guides and publications
- Tours, classes and workshops
- Reference and research services
 - [subject librarians](#)
 - for non-Monash clients ([MONINFO](#))

Service QuickFind

Choose from...

Go!

Libraries

Opening hours

Shortcut to...

Go!

Services for...

Undergraduates

Postgraduates

Staff

Alumni

CAVAL

Visitors

Off-campus students

Students with disabilities

Send inquiries and comments to [e-Query](#)
Last updated 19 July 2001 - Today is 2 August 2001 06:15 PM

Figure 2.6 Monash University Library main page screen shot [9]

The Monash University Library web page contains information about:

1. Information about the services
2. Eligibility for the services
3. How to place a request
4. Delivery options
5. How long it takes to receive materials
6. Costs
7. Payment
8. Thesis
9. Copyright Regulations

Opinion about the website

The Monash University Library web page has the links, which are:

- i. Borrowing from Monash Library
- ii. CAVAL
- iii. Interlibrary Loan via Doc Del
- iv. Australian libraries
- v. Borrowing rights for visitors
- vi. Photocopying
- vii. Computer facilities
- viii. Study areas
- ix. Connecting to electronic databases and journals
- x. Electronic request
- xi. Flexible library services unit
- xii. Inquiries & Assistance

Figure 2.6 shows Monash University Library web page screen shot. The Basically the layout and combination of colors and fonts are formal, same as most of the approach

taken by other libraries. The unique feature of Monash Library web page is providing a search features for the users to search for services keywords. It also has a special column which consist of services available for specific group of users which are:

1. Undergraduates
2. Postgraduates
3. Staff
4. Alumni
5. CAVAL
6. Visitors
7. Off-campus students
8. Students with disabilities

This features enables different group of users to viewed the specific or related services rather than looking at all of the available services although users are not eligible in.

7. SHPL

URL : <http://www.lib.monash.edu.au/docdel>

Date accessed : 30 July 2001

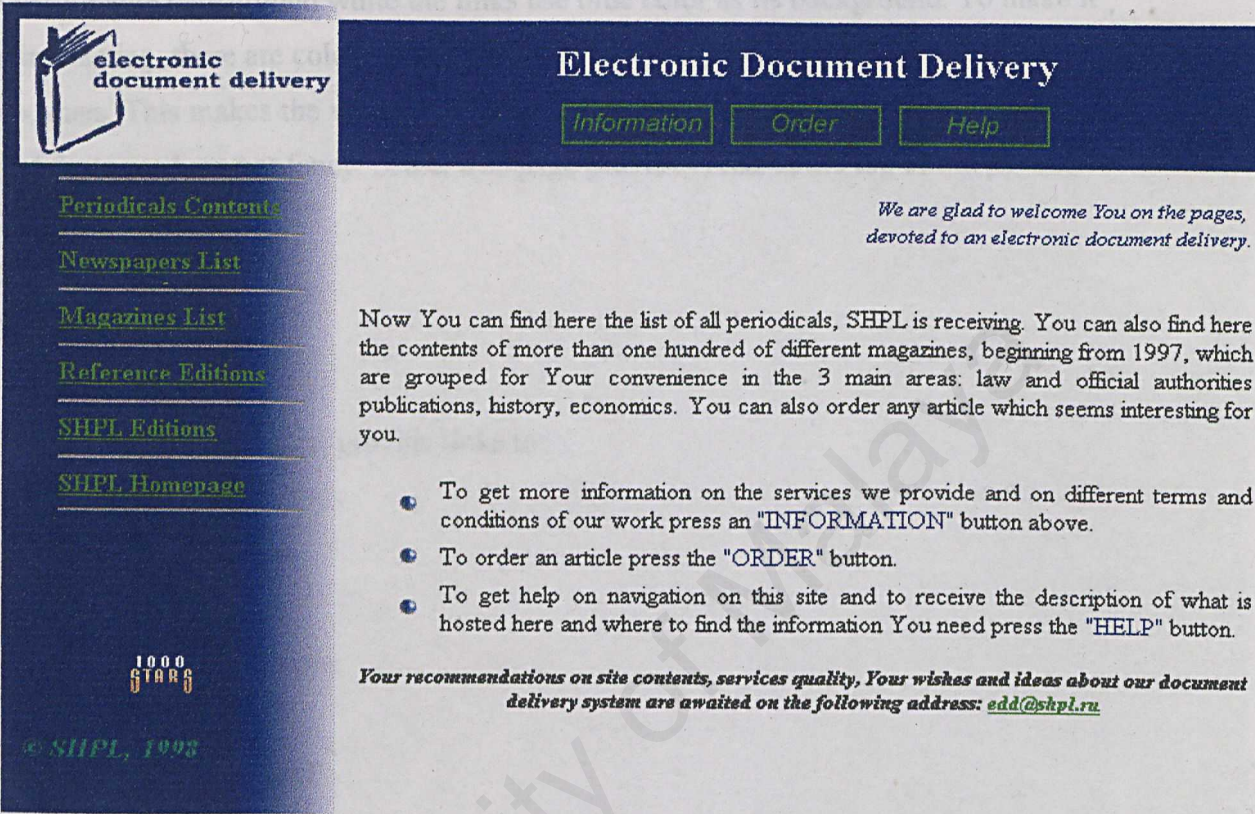


Figure 2.7 SHPL main page screen shot [10]

Figure 2.7 shows the SHPL main page screen shot. The SHPL web page contains information about:

1. Information about the services
2. How to place a request
3. Costs
4. Type of materials that can be requested
5. How long it takes to receive materials
6. Payment

Opinion about the website

SHPL uses a combination of simple colors to attract users attention. The main body of the page has a white background while the links use blue color as its background. To make it more interesting, there are color gradient between the white and blue background, instead of two tones. This makes the web site look livelier rather than flat. The types, colors and sizes of fonts used are not fancy. SHPL web page provides links at the top of the page to:

1. Information
2. Order
3. Help

While the right hand side, provide links to:

1. Periodical Contents
2. Newspaper List
3. Magazines List
4. Reference Editions
5. SHPL Editions
6. SHPL Homepage

Basically, these links are about the type of materials that can be requested or provided by SHPL.

8. MUSC

URL : <http://www.library.musc.edu/home.html>

Date accessed : 30 July 2001

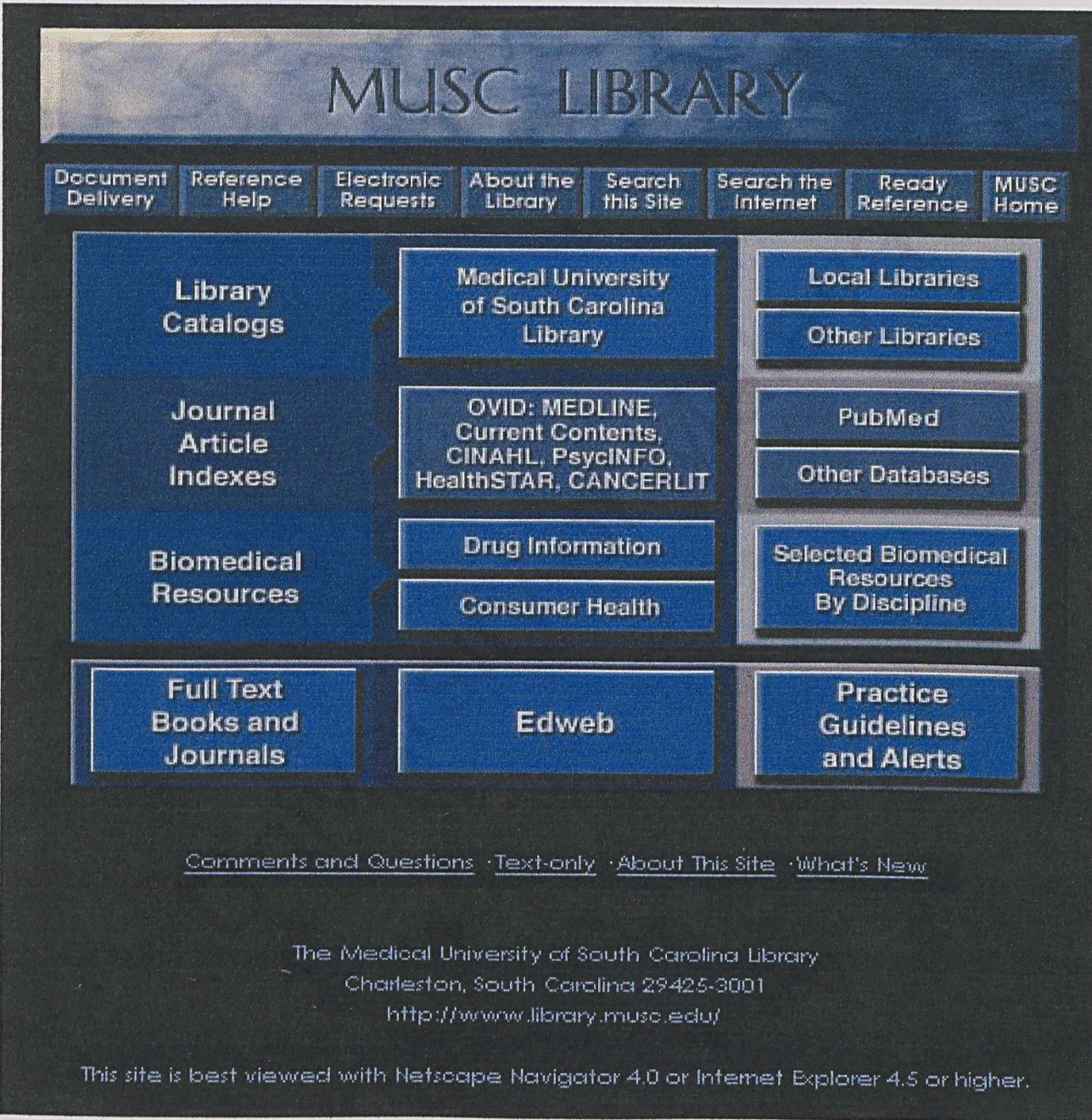


Figure 2.8 MUSC Document Delivery Services screen shot [11]

Figure 2.8 shows MUSC Document Delivery Services web page screen shot.

The SHPL web page contains information about:

1. Information about the services
2. How to place a request
3. Costs
4. Type of materials that can be requested
5. How long it takes to receive materials
6. Payment

Opinion about the website

MUSC uses a combination of stylish colors to attract users attention. The main body of the page has a black background while the links use blue color as its background. The links and services are organized in a tabular form, which make it look neat. Although the style of the MUSC web page is stylish, the type of colors and sizes of fonts used rather formal. MUSC web page provides links at the top of the page to:

1. Document delivery
2. Reference help
3. Electronic request
4. About the library
5. Search the site
6. Search the Internet
7. Ready reference
8. MUSC home

9. National Library of Australia

URL : <http://www.nla.gov.au/dss/loansome.html>

Date accessed : 30 July 2001

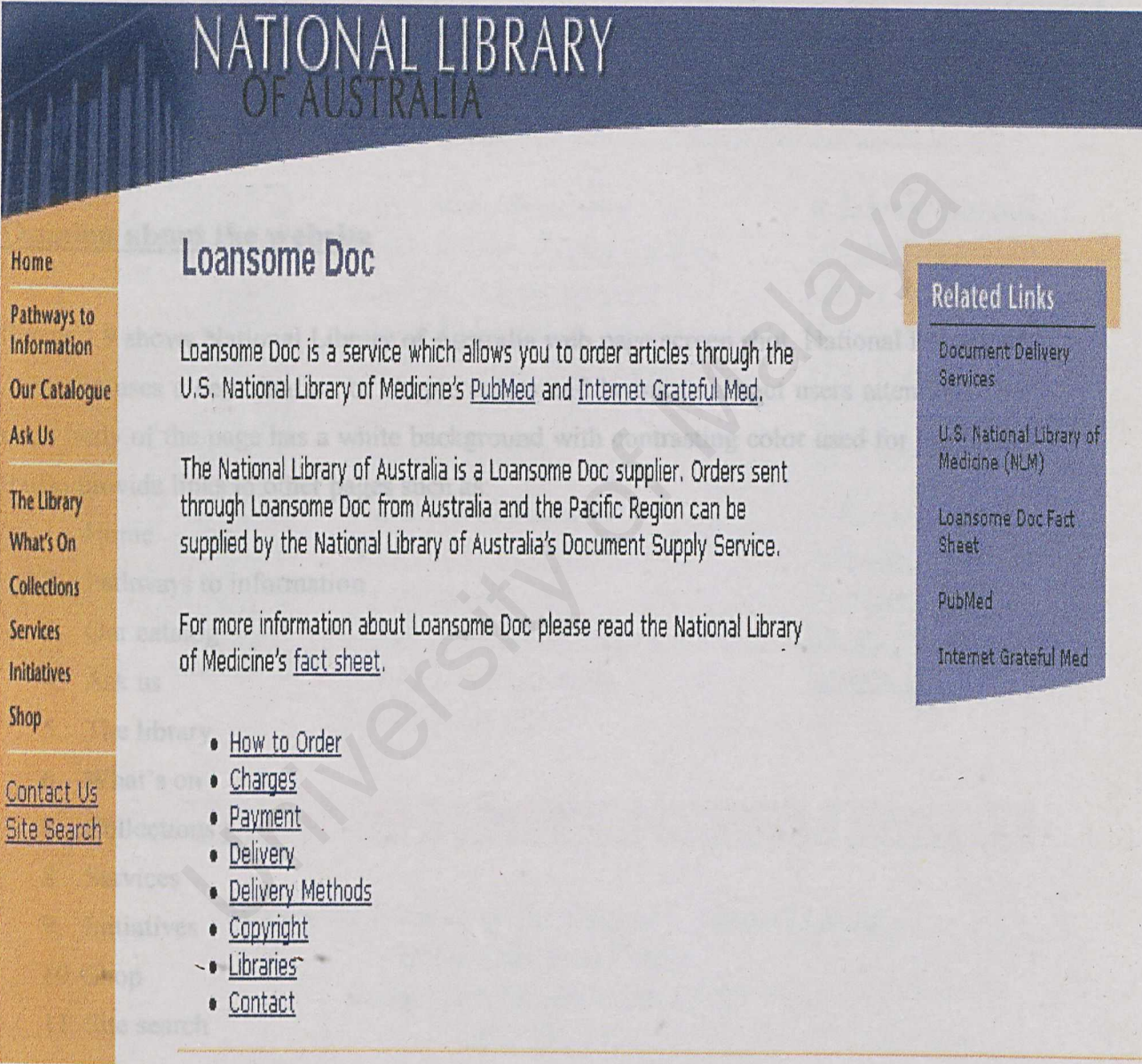


Figure 2.9 National Library of Australia Loansome Doc screen shot [12]

The National Library of Australia web page contains information about:

1. How to place a request
2. Costs
3. Payment
4. Delivery option
5. How long it takes to receive materials
6. Copyright regulations
7. Contact

Opinion about the website

Figure 2.9 shows National Library of Australia web page screen shot. National Library of Australia uses a combination of simple colors and layout to attract users attention. The main body of the page has a white background with contrasting color used for the fonts.

It also provide links to other pages such as:

1. Home
2. Pathways to information
3. Our catalog
4. Ask us
5. The library
6. What's on
7. Collections
8. Services
9. Initiatives
10. Shop
11. Site search

Besides than the links listed above, there is also a related links located on the right of the web page. All the information is organized neatly. They use formal fonts type and size to deliver messages to their users.

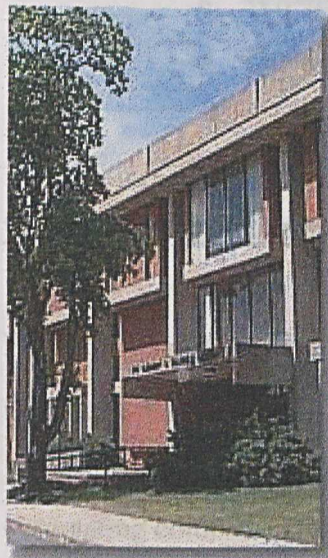
10. The Clement C. Maxwell Library

URL : <http://www.bridgew.edu/depts/maxwell/library.htm>

Date accessed : 30 July 2001

BRIDGEWATER STATE COLLEGE HOME

The Clement C. Maxwell Library



• *Electronic Resources*

[Search Webster \(Online Catalog\)](#)
[Databases & Search Guides](#)
[Internet Resources by Subject](#)
[Full-Text Journal Title Search](#)
[Online Indexes & Abstracts](#)
[Electronic Reserve](#)

• *Library Collections*

[Library Resource Guides](#)
[Periodical List](#)
[Recent Acquisitions](#)
[Library Liaison Program](#)
[Faculty Publications](#)

• *Library Services*

[Library Hours](#)
[Library Service Units](#)
[Computer Lab](#)
[Floor Plans](#)
[Circulation/Reserves](#)
[Off-Campus Support](#)

• *Electronic Forms*

[Acquisitions Request](#)
[Document Delivery](#)
[Library Instruction](#)
[Reserve Request](#)

What's New?

FAQ

Employment

Staff Directory

Comments

This page is maintained by The Clement C. Maxwell Library

Bridgewater State College

Bridgewater, Massachusetts 02325

(508) 531-1392

Figure 2.10 The Clement C. Maxwell Library screen shot [13]

2. The Clement C. Maxwell Library web page contains information about:

1. Information about the services
2. Eligibility for the services
3. Type of materials can be requested
4. How long it takes to receive materials
5. How to place a request
6. Costs
7. Checkout period
8. Where to pick up the materials
9. Further enquiries

Opinion about the website

Figure 2.10 shows The Clement C. Maxwell Library web page screen shot. The Clement C. Maxwell Library uses a combination of simple colors and layout to attract users attention. The main body of the page has a white background with contrasting color used for the fonts. It also provide links to other pages such as:

1. Electronic resources
2. Library services
3. Library collections
4. Electronic forms

Besides than the links listed above, there is also a related links located on the bottom of the web page, which are what's new, FAC, employment, staff directory and comments. The above mention links are organized in a tabular form, it organizes the category of links into block of links. The links at the bottom of the web page are organized in a line.

2.9 Summary of web sites reviewed

There are few areas which the system that will be build are interested in, that are:

1. Type of materials can be requested
2. Way to request
3. Notification
4. Delivery method
5. Delivery format for materials

1. Type of materials can be requested

Types of materials that can be requested using this service are

- a. Books
- b. Journal articles
- c. Theses
- d. Dissertations
- e. Government document
- f. Microfilm or fiche
- g. Patents
- h. Periodical articles
- i. Electronic documents

These types of materials can be only requested through the service only if they are not available in the institution.

2. Way to request

Different libraries have different type of request method but typically all users must fill up a request form first in order to place an order.

Monash University Library and RMIT University Library for example, request user to register as a user using the registration form available online before they are eligible to use the services. Other libraries does not require user to specially register as the system user, instead they can use their institution ID to log in request.

The requests form for Syracuse University Library, Stanford University Library, RMIT Library and Monash University Library are available online. The user needs to fill up an e-form instead of a printed form. University of Nevada do offer the request for services through printed form available at their document delivery office. Another approach can be seen is having a printable version form online so that user can print out the form and fill up the form before they can submit to the library.

For Stanford University Library, a separate form is available for different request type such as request for book form and request for article form.

3. Notification

E-mail, telephone or mail is the normal way to notify the user when the materials requested are ready for collection.

4. Delivery method

For Stanford University Library, photocopied materials will be mail to the address provided by the user. Books must be picked up in the Library Services Office, microforms must be used in the Media-Microtext room and some items may only be used in the Special Collections reading room.

The Nevada University takes different approach. The materials may be picked up at the Copy Center or the materials may be mail using campus mail, local mail and fax to user with a pre-payment made.

For RMIT University Library, the approach taken can be seen to have some typical method and some unique method. All the photocopies materials are for user to keep and for loan items, user must return the materials by the due date. RMIT uses the Internet to mail the materials to user's department or uses fax, mail or even email to deliver the materials. Note that email only applicable for materials available in electronic form. For loan materials, user will need to collect the materials upon arrival at the library itself.

The approach taken by RMIT University Library is typical with the approach taken by the Monash University Library.

5. Delivery format for materials

All the materials delivered to the user are in form of hardcopy. Photocopies materials are for the user to keep. Electronic document on the other hand are mostly in pdf format and will be send to the user via email.

6. Web page features

Table 2.1 shows the summary of the web pages analyzed features.

Institution	Links	Download time	Fonts & Size	Color	Layout	Multi-media	Service search
University of Nevada Libraries Library	Yes	Average	Formal	Normal	Formal	No	No
Syracuse University Library	Yes	Average	Formal	Normal	Formal	No	No
Stanford University Library	Yes	Average	Formal	Normal	Formal	No	No
Arizona State University Libraries	Yes	Average	Formal	Normal	Formal	No	No
RMIT University Library	Yes	Slow	Stylish	Stylish	Stylish	Yes	No
Monash University Library	Yes	Average	Formal	Normal	Formal	No	Yes
SHPL	Yes	Average	Formal	Normal	Formal	No	No
MUSC Library	Yes	Average	Formal	Normal	Stylish	No	No
National Library of Australia	Yes	Average	Formal	Normal	Formal	No	Yes
The Clement C. Maxwell Library	Yes	Average	Formal	Normal	Formal	No	No

Table 2.1 Web page features summary table

2.10 Sample of request forms

Samples of request forms are all in electronic form. We are more interested in the electronic forms because the system that will be developed is a web-based system. Some of the forms are available in printed copies which user can download and print the form them selves. Some universities library do have separate form for requesting materials such as form for books, form for articles and form for a chapter of a book.

Registration form

Firstly is about the registration form. Some libraries require their user to first register as a user for the system before they can actually use the services. Filling up the electronic registration form available on the library web site can do this. After completion, the user will be given a password so they can proceed by making request using that password. Other libraries will require their users to key in their institution student ID as access to making a request. Figure 2.11 shows the Arizona State University Libraries Document Delivery Services registration form.

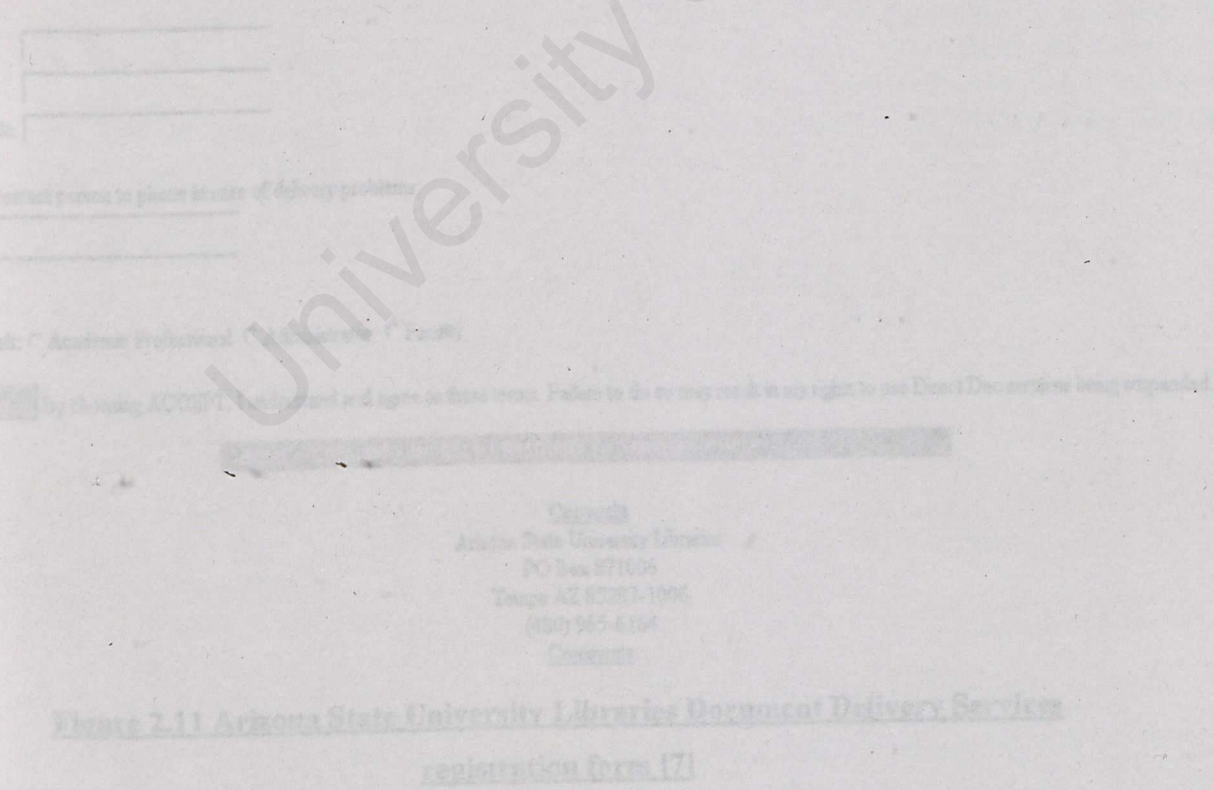


Figure 2.11 Arizona State University Libraries Document Delivery Services registration form [7]

ARIZONA STATE UNIVERSITY LIBRARIES

[HOME](#) • [CATALOG](#) • [INDEXES](#) • [RESOURCES](#) • [SERVICES](#) • [HELP](#)

Arizona State University

Direct Doc Services Registration Form

Direct Doc funding supports research, instruction, and administrative endeavors of faculty, academic professionals and administrators on the ASU Main Campus. Staff (graduate, office, research or teaching assistants, etc.) may be trained to perform the activities, but all orders and delivery instructions are the responsibility of the registered participant and must be made in that person's name. Passwords are NOT to be assigned to students for use to fulfill curricular or assignments.

ID #:
Department:
Address:
Number:
Number:

Main Campus Address To Which You Wish Paper Documents Delivered

City:
State:
Zip:

Contact person to phone in case of delivery problems

Rank: ☐ Academic Professional ☐ Administrator ☐ Faculty

☐ **ACCEPT** By choosing ACCEPT, I understand and agree to these terms. Failure to do so may result in my rights to use Direct Doc services being suspended.

[HOME](#) • [CATALOG](#) • [INDEXES](#) • [RESOURCES](#) • [SERVICES](#) • [HELP](#)

[Copyright](#)
Arizona State University Libraries
PO Box 871006
Tempe AZ 85287-1006
(480) 965-6164
[Comments](#)

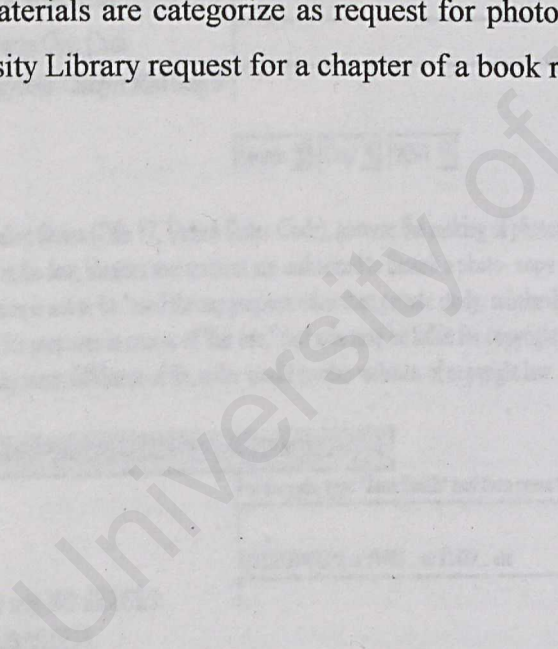
**Figure 2.11 Arizona State University Libraries Document Delivery Services
registration form [7]**

Request for materials form

Request for materials forms are separated to different kinds of request such as:

- 1. Request for a book
- 2. Request for an article
- 3. Request for chapters of book
- 4. Request for dissertation or thesis
- 5. Request for conference papers
- 6. Request for government publication

University of Nevada Libraries takes this approach. The way they separate the forms will make it more clearly to the user in the way the request is categorize. All request for materials except books, conference proceedings and newspapers on microfilm, and any other library materials are categorize as request for photocopies. Figure 2.12 shows the Nevada University Library request for a chapter of a book request form.



Electronic Document Delivery System

Chapter 2 – Literature Review

Library Catalog for Northern Nevada Academic Libraries

[Start Over](#) | [\(Another ILL Request\)](#)

Use this form to request photocopies of a chapter in a book.

Chapter Author	<input type="text"/>
Chapter Title	<input type="text"/>
Title of Book	<input type="text"/>
Book Author	<input type="text"/>
Pages in Book	<input type="text"/>
Publisher & Date	<input type="text"/>
Source of Reference	<input type="text"/>
Other Information: Missing, at WNCC, etc.	<input type="text"/>
Payment Options: DPO, Charge Circ, Cash	<input type="text"/>
Delivery Options with Prepayment: Campus Mail Stop #	<input type="text"/>
or Email address	<input type="text"/>
Cancel if not filled by	<input type="text"/>

The copyright law of the United States (Title 17, United States Code), governs the making of photocopies or other reproductions of copyrighted material. Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use," that user may be liable for copyright infringement. This institution reserves the right to refuse to accept a copying order, if, in its judgement, fulfillment of the order would involve violation of copyright law.

SEARCH THE LIBRARY CATALOG FOR THIS ITEM

For example, type "Jane Smith" and then press the RETURN key.

Your Name:	<input type="text"/>
Type your barcode number with NO SPACES:	<input type="text"/>

[SUBMIT THIS REQUEST](#)

- [Interlibrary Loan Book Request](#)
- [Interlibrary Loan Chapter Request](#)
- [Interlibrary Loan Dissertation Request](#)
- [Interlibrary Loan Government Document Request](#)
- [Interlibrary Loan Journal Request](#)
- [Interlibrary Loan Conference Proceedings Request](#)

Figure 2.12 Nevada University Library request for a chapter of a book request form [4]

Electronic Document Delivery System
Chapter 2 – Literature Review

As for electronic document delivery services, the forms are available in electronic form. The request for materials is also categorize as request for a book and also request for an article as steps taken by the Medical University of South Carolina Library. Figure 2.13 shows the MUSC Document Delivery web page showing categorize forms.

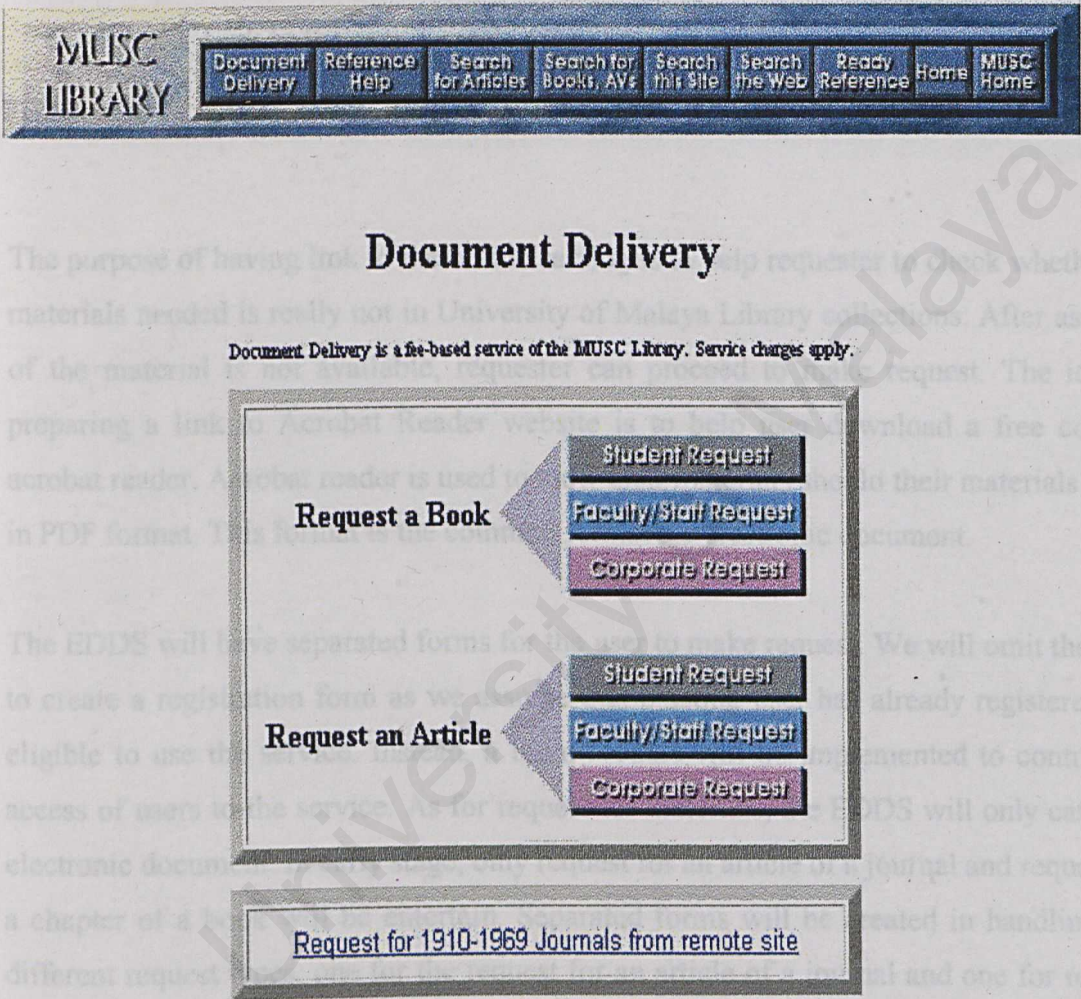


Figure 2.13 MUSC Document Delivery categorize forms web page [11].

2.11 Summary

From what have been review and analysis, the suggested EDDS will have some features which most of the existing have. The features, which the EDDS will have, are:

1. Links to library main web page
2. Link to other library web pages
3. Link to the online catalog
4. Link to the Acrobat Reader website.

The purpose of having link to the online catalog is to help requester to check whether the materials needed is really not in University of Malaya Library collections. After assuring of the material is not available, requester can proceed to make request. The idea of preparing a link to Acrobat Reader website is to help user download a free copy of acrobat reader. Acrobat reader is used to view their materials should their materials arrive in PDF format. This format is the common format for electronic document.

The EDDS will have separated forms for the user to make request. We will omit the need to create a registration form as we assume that existing user has already registered and eligible to use the service. Instead, a login module will be implemented to control the access of users to the service. As for request for materials, the EDDS will only cater for electronic document. In early stage, only request for an article of a journal and request for a chapter of a book will be entertain. Separated forms will be created in handling two different request types, one for the request for an article of a journal and one for request for a chapter of a book. It is advisable because the types of information needed to proceed with processing the request are different and unique.

Furthermore, enquiries and suggestions services will be provided to help user in understanding how to use the system should they have doubt and to help improve the service and the features of the system.

All the above mentioned features and suggestions are being made after analyzing existing systems and relevant systems to suit the need for University of Malaya Library. In other words, the system will be localized to suit our need and environment.

The EDDS system will be a web-based system, same as the reviewed systems. In developing the EDDS, web-authoring tools such as HTML and ASP will be used because the web pages that will be developed are dynamic pages. This is because the web pages need to capture users' information. A database management system (DBMS) will also be needed to manage all the data captured through the electronic forms and user interfaces.

Methodology and
System Analysis
University of Malaya

3.1 Methodology

Managing software engineering projects is tough. The body of methods, rules, postulates, procedures, and processes that are used to manage a software engineering project are collectively referred to as methodology [14]. This will ensure a proper documentation on the work and tasks need to be carried out. Waterfall methodology will be used to develop this system.

Chapter 3

3.1.1 Waterfall

There are four main phases in the Waterfall Methodology [15]

1. Analysis phase
2. Design phase

Methodology and System Analysis

This methodology is so called Waterfall methodology because each phase flows naturally into the next phase. This methodology is one of the most powerful methodologies. It states what one should think first what is being done before establishing the plan to study and show how it should be built and finally build it. [17]

It allows for a software engineering methodology, which is more in alignment with hardware engineering methods and practices. It also forces discipline process to avoid pressures of writing code long before it is known what is to be built. [17]

In many cases, codes are being built and later to be known that it was needed or will be in little use to the product. These codes becomes legacy and too costly to abandon. This methodology has the solution to overcome this problem. This is made possible by forcing analysis and planning before actions are taken. [17]

3.1 Methodology

Managing software engineering projects is tough. The body of methods, rules, postulates, procedures, and processes that are used to manage a software engineering project are collectively referred to as methodology [14]. This will ensure a proper documentation on the work and tasks need to be carried out. Waterfall Methodology will be used to develop this system.

3.1.1 Waterfall Methodology

There are four main phases in the Waterfall Methodology [15].

1. Analysis phase
2. Design phase
3. Implementation phase
4. Testing phase

This methodology is so called Waterfall methodology because each phase flows naturally into the next phase like water over a series of falls. Figure 3.1 show the diagram of Waterfall Methodology. [16]

This methodology is one of the very powerful methodologies. It states what one should think first what is being built before establishing the plan to study and show how it should be built and finally built it. [17]

It allows for a software engineering methodology, which is more in alignment with hardware engineering methods and practices. It also forces discipline process to avoid pressures of writing code long before it is known what is to be built. [17]

Many cases, codes are being built and later to be known that it was needed or will be in little use to the product. These codes becomes legacy and too costly to abandon. This methodology has the solution to overcome this problem. This is made possible by forcing analysis and planning before actions are taken. [17]

Electronic Document Delivery System
Chapter 3 – Methodology and System Analysis

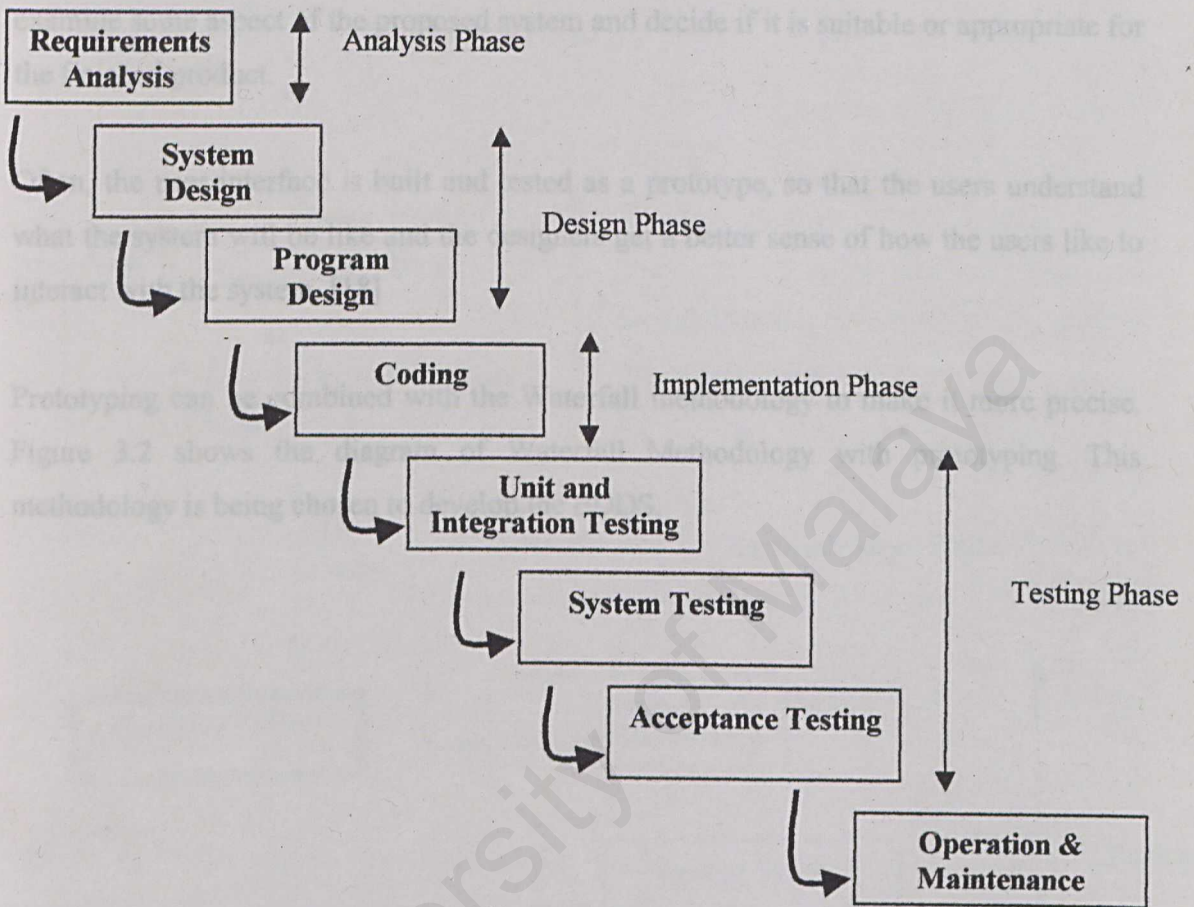


Figure 3.1: The waterfall model [18]

3.1.2 Prototype

Prototype is partially developed product that enables customers and developers to examine some aspect of the proposed system and decide if it is suitable or appropriate for the finished product.

Often, the user interface is built and tested as a prototype, so that the users understand what the system will be like and the designers get a better sense of how the users like to interact with the system. [18]

Prototyping can be combined with the Waterfall methodology to make it more precise. Figure 3.2 shows the diagram of Waterfall Methodology with prototyping. This methodology is being chosen to develop the EDDS.

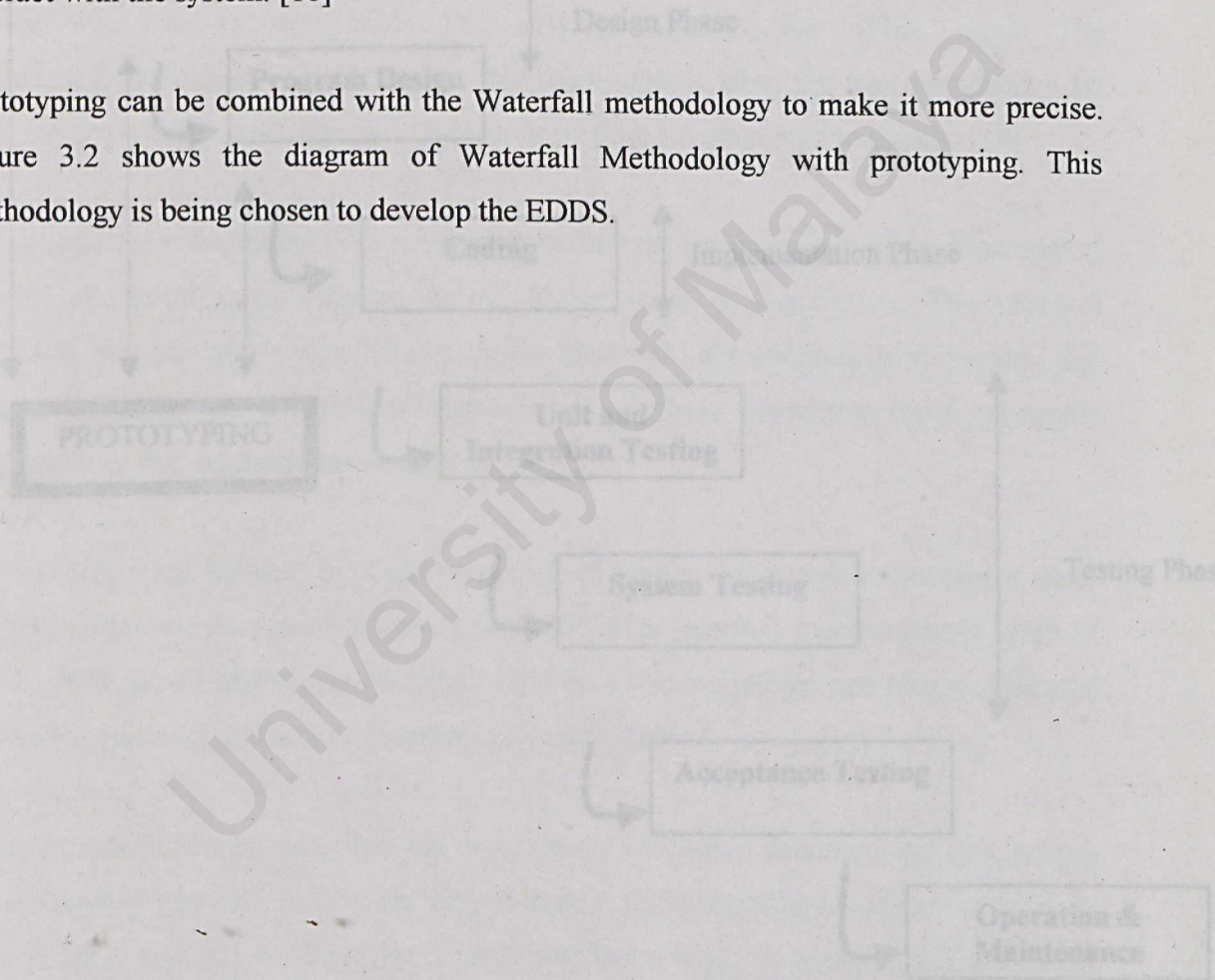
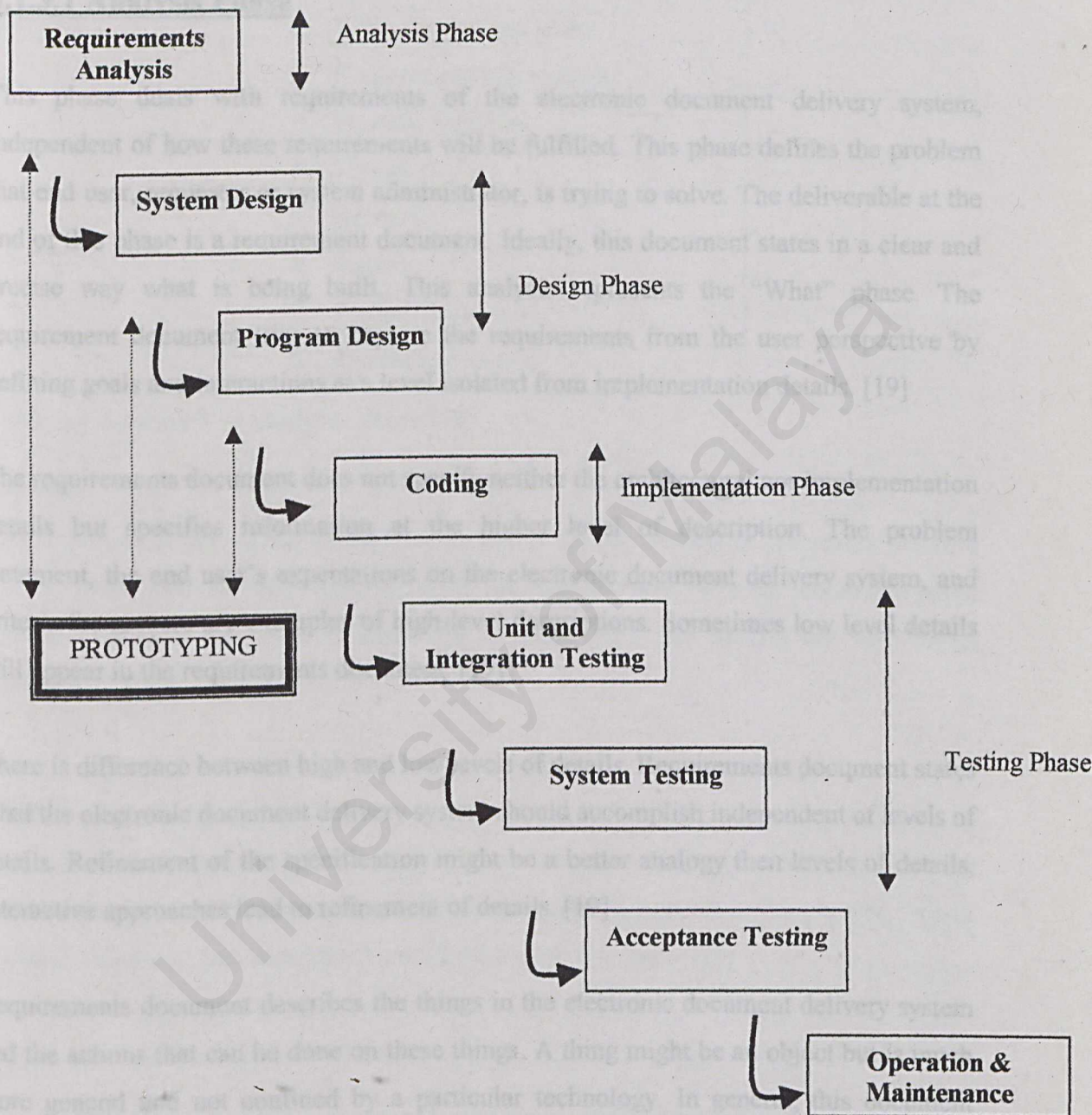


Figure 3.2: The waterfall model with prototyping. [18]



3.1.3 Activities in each phases of Waterfall Methodology

3.1.3.1 Analysis Phase

This phase deals with requirements of the electronic document delivery system, independent of how these requirements will be fulfilled. This phase defines the problem that end user, requester or system administrator, is trying to solve. The deliverable at the end of this phase is a requirement document. Ideally, this document states in a clear and precise way what is being built. This analysis represents the “What” phase. The requirement document tries to capture the requirements from the user perspective by defining goals and interactions at a level isolated from implementation details. [19]

The requirements document does not specify neither the architectural nor implementation details but specifies information at the higher level of description. The problem statement, the end user’s expectations on the electronic document delivery system, and criteria for success are examples of high-level descriptions. Sometimes low level details will appear in the requirements document. [19]

There is difference between high and low levels of details. Requirements document states what the electronic document delivery system should accomplish independent of levels of details. Refinement of the specification might be a better analogy then levels of details. Interactive approaches lead to refinement of details. [19]

Requirements document describes the things in the electronic document delivery system and the actions that can be done on these things. A thing might be an object but is much more general and not confined by a particular technology. In general, this document describes ontology, that is the noun phrases and the verb phrases that will become the guidelines for the defining application specific protocol. The decomposition of the problem is very important leading to the development of data structures and algorithms. [19]

Summary of Analysis Phase

Table 3.1 shows the summary of activities in analysis phase.

Phase	Deliverable
Analysis	<div>1. What do we want to do!</div> <div>2. Requirements Document</div> <div>3. Domain Ontology</div> <div>4. Things</div> <div>5. Actions</div> <div>6. States</div> <div>7. Typical Scenarios</div> <div>8. Atypical Scenarios</div>

Table 3.1 Summary of analysis phase [19]

Things

The requirement document defines the ontology of the electronic document delivery system, which is in the more general sense, the noun phrases. Here the pieces and parts, constants, names, and their relationships to each other are among the few things specified. [19]

Actions

The requirement document defines the actions that the electronic document delivery system can perform. This is expressed, in the more general sense, as verb phrases. Object methods, functions, and procedures are all examples of actions. [19]

States

State is defined as a sequence of settings and values, which distinguishes one time-space slice of a system from another slice. Every system, unless it is truly stateless, goes through a series of states. Among them could be the initial and final state. Error states might also be detected and defined. Most of the states are domain specific. [19]

Typical Scenarios

A scenario is a sequence of steps taken to accomplish a given goal. When the electronic document delivery system is completed and the application is available, the end user should be able, in an easy and clearly specified manner, accomplish all typical usage, or scenarios, for the application. The typical scenarios represent the majority of uses for the electronic document delivery system. [19]

Atypical Scenarios

An atypical scenario is something that needs to be accomplished with the electronic document delivery system, but only seldom. The actions have to be done correctly, but perhaps, at lower efficiency. One would hope that an unexpected error condition is an atypical event. Nonetheless, the electronic document delivery system should be able to deal with many categories of faults by using several established techniques such as exception handlers, replications, and process monitoring and roll over. [19]

Incomplete and Non-Monotonic Requirements

The entire enumeration of all of the requirements is not possible for nearly all real-life situations. Godel's incompleteness theorem of arithmetic says that there is no finite list of axioms that completely describe integer arithmetic. Expressed in our terminology, there is no finite list of requirements that would completely describe arithmetic. Since integer arithmetic is an underlying foundation of most computer hardware systems and software applications, and since we can't even enumerate the requirements for integer arithmetic, the task of enumerating a more complex system completely is untraceable. [19]

The requirement gathering process is iterative and non-monotonic in nature. Early on in the process, some requirements are established. As the process continues, other new requirements are discovered which may be in conflict with earlier known requirements, thus leading to a new structure. [19]

In traditional logic a theory is defined by a finite set of axioms. Theorems in the theory are valid sentences. If new axioms are added to the theory, the already existing theorems remain valid and the theory is extended into a new theory with new theorems added to the established theorems. [19]

In non-monotonic logic added new axioms to the theory may invalidate existing theorems. A new theory is created which are not a simple extension of the old theory but a collection of new theorems and some of the established theorems. [19]

The requirement process is more like non-monotonic logic than monotonic logic. An initial collection of requirements, the axioms of the system, defines the capabilities, the theorems of the system. New requirements may lead to a collection of capabilities entirely different than the established capabilities. New requirements may negate old solutions. [19]

There is hope for requirement gathering. The seven Peano axioms of integer arithmetic correctly describe an infinite cardinality of integers. Only the integers that are beyond infinity fall in the collection of under specified integers. If we are willing to limit ourselves to an infinite number of integers that are less than infinity, we can establish the requirements for integer arithmetic in a finite manor. Though the requirements may not theoretically describe that system, the coverage may be quite satisfactory. [19]

3.1.3.2 Design Phase

This phase, the architecture of the electronic document delivery system is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture. Architecture defines the electronic document delivery system components, their interfaces and behaviors. The deliverable design document is the architecture. It describes a plan to implement requirements. This phase represents the “How” phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces and many other engineering details are established. [20]

The information established in the requirement document will be expanded. Using typical and atypical scenarios provided from the requirement document, performance tradeoffs can be accomplished as well as complexity of implementation tradeoffs. Analyzing tradeoffs of necessary complexity allows for many things to remain simple, which in turn, will eventually lead to higher quality product. [20]

The typical scenarios are converted into a test plan. Given a complete requirement document, critical priorities must be indicated. A critical implementation priority leads to a task that has to be done right. If it fails, the product fails. If it succeeds, the product might succeed. [20]

Summary of Design Phase

Table 3.2 shows the summary of activities involved in the design phase.

Phase	Deliverable
Design	<div><div>1. What is our plan?</div><div>2. Architecture Document</div><div>3. Implementation plan</div><div>4. Critical Priority Analysis</div><div>5. Performance Analysis</div><div>6. Test Plan</div></div>

Table 3.2 Summary of design phase [20]

Architecture

The architecture defines the electronic document delivery system components, their interfaces and behaviors. [20]

Implementation Plan

The implementation plan establishes the schedule and needed resources to develop the electronic document delivery system. It defines implementation details including programming languages, platforms, programming environments, debuggers, and many more. [20]

Critical Priority Analysis

The critical analysis generates a list of critical tasks. It is absolutely necessary to successfully accomplish a critical task. The electronic document delivery system project will succeed or fail based on the outcome of these tasks. Some projects may have more than one critical task. There are two major categories of critical tasks. One category of tasks is associated with the building of the electronic document delivery system. The other category of critical tasks is associated with the system itself. It is absolutely necessary to successfully accomplish both categories of critical tasks. [20]

Performance Analysis

Once given the typical scenarios from the requirement document, the electronic document delivery system can be designed to meet performance objectives. [20]

Test Plan

The test plan defines the gold standard. If the electronic document delivery system passes all tests in the test plan, then it is declared to be complete and of high quality. The more complete the coverage of the system is, the higher the confidence in the system and the system's quality rises. [20]

3.1.3.3 Implementation Phase

This is the stage where the electronic document delivery system’s components are built. Given the document from the design phase and the requirements document from the analysis phase, components should be built exactly what is requested though there is still room for flexibility. For example, a component may be special cased for only this particular system, or the component may be made more general to satisfy reusability guideline across many systems forming a product line. The requirement document or architecture document should give guidance. [21]

The implementation stage deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself. [21]

Summary of Implementation Phase

Table 3.3 shows the summary of activities involved in the implementation phase.

Phase	Deliverable
Implementation	1. Now built it! 2. Code 3. Critical Error Removal

Table 3.3 Summary of implementation phase [21]

Critical Error Removal

Critical error is an error which prevents the electronic document delivery system from fully satisfying the usage scenarios. There may indeed be many known errors in the product. Usually these errors are listed in the release notes and have well-established work around. In fact, the system may have, yet-to-be-discovered errors. [21]

3.1.3.4. Testing Phase

This stage exist because of the statement “quality is very important”. [22]

In methodologies, testing phase is separated phase, which is performed after the implementation is completed. There is a merit in this approach because it is hard to see own (designer and programmer) mistakes and a fresh eye can discover obvious errors much faster than the person who has read and re-read the materials many times. [22]

Regardless if testing is done after-the-fact, or continuously, testing is usually based on a regression techniques, split into several major focuses consisting of internal, unit, application and stress. [22]

Summary of Testing Phase

Table 3.4 shows the summary of activities involved in the testing phase.

Phase	Deliverable
Testing	<div>1. Improve quality</div> <div>2. Regression Test</div> <div>3. Internal Testing</div> <div>4. Unit Testing</div> <div>5. Application Testing</div> <div>6. Stress Testing</div>

Table 3.4 Summary of testing phase [22]

The next focus deals with testing a unit as whole. This would test the interaction of many functions but confine the user within one unit. The exact scope of a unit is left to interpretation. Supporting test code might be necessary to support an individual test. The architecture and implementation teams drive this type of testing. [22]

Regression Test

Quality is usually based on a suite of regression tests. A regression test is written and the results are generated. If the results are in error, then the offending bug s corrected. Finally, the regression test generates verified results. These verified results are called the gold standard. The terminology “gold standard” was borrowed from financial markets where paper money issued by governments was backed by gold. [22]

Ideally, the validity of a test results would be driven by the requirements document but in practice the implementation team makes this interpretation. The tests are collected, as well as their gold standard results, into regression test suite. As development continues, more tests are added, while old test must still remain valid. The test suite is run generating new results. These new results are then compared with the gold standard results. If they differ, then a potential new fault has entered the system. The fault is corrected and the development continues. The mechanism detects when new development invalidates existing development and thus prevents the system from regressing into fault state. [22]

There are four major focuses of regression testing used to assure quality.

Internal Testing

The first focus deals with low-level implementation. Here each function or component is tested. The implementation team accomplishes this testing. [22]

Unit Testing

The next focus deals with testing a unit as whole. This would test the interaction of many functions but confine the test within one unit. The exact scope of a unit is left to interpretation. Supporting test code might be necessary to support an individual test. The architecture and implementation teams drive this type of testing. [22]

Application Testing

The next focus deals with tests for the entire application. The scenarios from the analysis team drive the testing. [22]

Stress Testing

The next focus of testing, deals with application in the environment. This category is called stress testing. The idea is to create an environment more demanding of the application would experience under normal workloads. Race conditions and memory leaks are often found under stress testing. This is most complex and hardest category of testing to accomplish and requires a joint effort of all teams. [22]

Table 3.5 shows the type of testing and their corresponding task and also the people involved in the testing. [22]

Focus	Note	Team responsibility
Internal	Make sure all internal, non user-visible components work well.	Implementation Team
Unit	Make sure all user visible components work well.	Implementation and Design Teams
Application	Make sure the application can complete all scenarios	Analysis Team
Stress	Run application in an environment that is more stressful than the target environment.	All team

Table 3.5 Types of testing and the corresponding action and responsible team for each action [22]

The types of testing that will be carried out in the testing of the electronic document delivery system are:

1. Internal testing
2. Unit testing
3. Application testing

Stress testing will not be implemented because it needs a lot of hardware equipment, tester and time.

3.2 Methods of collecting data

The methods used in collecting data required for developing electronic document delivery system are:

1. Searching the Internet
2. Interview
3. Discussion
4. Document analysis
5. Observation on available systems and related system

Searching the Internet

The Internet is a platform where a lot of information can be acquired. With development of search engines such as Googles, Copernic, Excite, Yahoo and Search, relevant information sites can be viewed with only a click away provided users key in the related keywords for search. From the search, there is no electronic document delivery services available locally being fully implemented yet. But there are many relevant and available systems implemented by oversea institutions. The keywords used in searching relevant web pages are:

1. Electronic Document Delivery System
2. Library Document Delivery Services
3. Library Electronic Document Delivery Services

The keywords for number 2 and 3 uses the “library” because it is widely known that libraries do offer and implement this kind of system and services.

Interview

An interview session has been conducted at the early stage of this project. The respondent of the interview is the University of Malaya Library’s officer at the enquiries counter. The purpose of the interview is to get a better view about document delivery services in University of Malaya. Sample of form also being received and would be used in helping the creating of the electronic form. From the interview also, the current scenarios and problems regarding the document delivery services is being revealed. The requirements for a much more effective way to organize the service is also being mentioned.

Discussion

Discussions were made with the adviser of the project and much of the details and idea on how to develop an electronic document delivery system. Students are also involved in the same discussions but in different session, and it is a fact to know that, most students do not know anything about the document delivery services or electronic document delivery services.

Document Analysis

Materials and writing about the electronic document delivery system and document delivery services are found by searching the Internet. There are many reviews about the electronic document delivery systems and being used to as research materials. Some of the development tips and ideas are captured and will be used as guideline in developing the system.

Observation on available and related system

Several systems and related systems found through the Internet. By observing the existing systems, ideas can be generated in helping the development process. But there are several restrictions and limitations about the systems found on the Internet.

1. The available systems are restricted to their registered users.
2. Most of the systems available are document delivery services and only a few systems actually implementing electronic document delivery services.
3. Available systems are located in overseas institutions and there is no systems found used locally.

3.3 Requirements Analysis

The requirements for the EDDS are categorized into two categories which are:

- i. Functional requirements
- ii. Non-functional requirements

3.3.1 Functional Requirements-

Below are the functional requirements of the system.

1. Users login

User must be a registered user before they can proceed with submission of request.

2. Making request

Users can make request using the specific form according to the type of materials they want to request

3. Submit request

When the user has confirmed all the details in the request form to be correct, user can click the submit button to submit their request.

4. Task list generations

When the submit button is clicked, all the information in the form will be captured and stored in the database. With the database management software, a task list may be generated as a reference to the librarian in helping them to locate materials for requester.

5. Summary report generation

Besides than generating a task list, the DBMS is also capable in generating reports for the system administrator and services executives. These reports can help them in planning the services well.

6. Notifying users upon materials arrival

Upon arrival of materials, librarian can contact the user to collect the by using email, telephone, fax or mail. These particulars are readily found in the database.

7. Online Help

Online help is available by clicking a link to send e-mail to the librarian.

8. Online Catalog Search

Online catalog search that is used to search materials in the system's host institution will be provided to help and prompt users to search and make sure that the materials needed are not located in the host's library.

3.3.2 Non-Functional Requirements

The non-functional requirements of the system are listed below.

1. Respond time

The response time to load the web pages and forms should be as fast as possible. Users will be de-motivated as they get stuck and wait for those web pages to be downloaded. The way to overcome this problem is to keep the web pages as simple as possible yet attractive, using smaller graphics file whenever graphics is needed only, avoid big animations features when it is not necessary and keeping the size of the whole web page as small as possible.

2. Security

Security in the system will be implemented by using users password and authentication manner. This is because each user has own unique ID.

3. User friendly

The system by itself should be made simple. Users will not need to know so much about how the systems goes but they should only be made to understand what they need to input when they want to make a request.

4. Multi-users environment

The system will be implemented as a web based system and services. Because it is a web-based system, there are possibilities that there will be multi-user trying to access and used the system at the same time frame. So it is important that the system will be able to handle this kind of situation.

5. Suggestions

Suggestions by the users or complains by the users can be logged by using email or suggestions box available at the service web page.

6. Web page design

Web page design should be kept simple as possible. A good combination of colors, fonts and size of fonts, pictures, links and layout is essential. Multimedia element should be incorporated when needed and the size should be kept as small as possible. This will affect the download time and response time of the web pages.

7. Links to other related pages

Links to the related sites and services will be provided to users. This is to help them navigate the web site easier.

8. Maintainability

The system can be corrected should an error occurred. Further more, it could adapt to new demand and requirements or enhanced in the future.

9. Expandability

The system should be able to be extended to accommodate more functionality in the future.

3.4 Selection of Programming Language and Development Tools

The programming language and tools that will be used to develop the system are:

1. **ASP 3.0**

ASP 3.0 is used to create active web pages.

2. **HTML 4.0**

HTML is used to create web pages. It is one of the elements in ASP.

3. **VB Script**

VB Script is a scripting language that will be used to create active codes. It is also one of the elements in ASP.

4. **Java Script**

JavaScript is also a scripting language. It will be the supporting scripting language besides VB Script. JavaScript support client side scripting and is supported by two major browsers, the Internet Explorer and Netscape Navigator.

5. **Notepad**

A non-graphical tool used to create Web pages. It is used for simple HTML editing.

6. **Visual Interdev 6.0**

A graphical tool used to create and manage web site. This will be the main tool used.

7. **Front Page 2000**

Also graphical tool used to facilitate the creation of web pages. Used as a supporting tool besides Visual Interdev.

3.5 System Requirements

3.5.1 Static and Dynamic Web Pages

3.5.1.1 Static Web page

Creating a web site in the early days of the World Wide Web (WWW) mainly concentrating on fancy graphics, nice looking and easy-to read features. They only display the information that is embedded in their HTML file and they do not do much of things except to deliver the message. The pages that can be seen on the user side, already been processed by the server and only the layout are displayed. These pages do not have any interactive features [23].

3.5.1.2 Dynamic Web Page

Nowadays, the function of a web page in the Internet is not only to display and deliver message to the user. Instead the Internet has made many things possible such as e-commerce activities that need to interact with users. This interaction includes searching for inventory, querying the database, making orders using shopping cart and also read others remarks on the page. This make the static web page are quickly becoming obsolete because they do not have the interactive features (which also refer to as dynamic web pages) [23].

3.5.2 Creating Dynamic Web Page

In the process of creating the system, Active Server Pages (abbreviated ASP) will be used to create the web pages. This is because the system will have to interact with user in the requesting process and to capture the users' information to be stored in the database [23].

3.5.2.1 Active Server Pages (ASP)

ASP is a solution by Microsoft to create dynamic web pages. The Active Server Pages are easy to use and easy to learn yet it promises great dynamic web pages as output. Active Server contains two segments, which are the pragmatic code and implanted HTML. Pragmatic code here refers to the scripting languages such as the VBScript, Jscript (supported by Internet Explorer for JavaScript), PerlScript and Python. But most ASP pages are created using the VBScript. The latest version of ASP is the ASP 3.0. ASP pages must have .ASP extension. ASP is based on server-side scripting technology [23].

3.5.2.2 VBScript

VBScript is similar to Visual Basic's syntax and it is very popular because many of the web developer have experience in using it. VBScript is one of the existing scripting languages, which has the syntax like English the most [23]. VBScript is an alternative to JavaScript in client-side scripting. The VBScript only supported by Microsoft Internet Explorer. In case for the Netscape web browser, they use the ScriptActive as plug-in to enable VBScript.

3.5.2.3 Java Script

JavaScript is a cross-platform scripting language, having the properties of simple, interpreted and object-oriented. An interactive web page can be produce by adding the scripting keywords into a HTML file. JavaScript started when LiveScript was developed by Netscape to provide an interface to Java and was rework by Sun. Internet Explorer 3.0 begin to give support to JavaScript through Jscript. The function of JavaScript in web pages are form validation, responding to input, detecting browser characteristics and integrating with Java basic graphics and dynamic HYML. Some of these functions are similar to the VBScript features. [23]

3.5.3 HTML

HTML is the abbreviation of Hypertext Markup Language. It is a client side language, which is supported by web browser. HTML is about how to use the HTML tags to produce web pages. HTML tags are the core element HTML. Different from the programs that we use, the code of the program is hidden from the user where else the HTML is just a click away from its user. User can actually view the HTML code to see what is going on. The latest version of HTML is HTML 4.0 and there are also other markup language such as XML and the latest XHTML, which is the combination of XML and HTML [24].

3.5.3.1 HTML 4.0

The latest HTML 4.0 uses some of the ideas and implement changes for HTML 3.0, which was replaced by HTML 3.2. HTML 4.0 allows separate physical styles from the content markup by relying more on style sheets. FRAME element is now formally defined outside the scope of Netscape and Microsoft. Furthermore, OBJECT element is introduced to address the competing interest between applets, plug-in, and ActiveX controls. [24]

The important features of HTML 4.0 includes [24]:

- i. HTML markup is separated from style through the introduction of style sheets
- ii. Added flexibility in handling forms and tables
- iii. Incorporation of scripting language capabilities
- iv. Ability to respond to or ignore specific media types
- v. Better printing capabilities
- vi. Improved sensitivity to different language types
- vii. End users can choose the style for viewing a document or turn off the style sheets features.

3.5.3.2 Cascading Style Sheet (CSS)

Cascading Style Sheet or CSS is an official name for the HTML style sheet tool]. The current version of CSS is version 2 (CSS2). However CSS version 1 is supported by most of the browser if compare with CSS2. CSS1 was a good, solid shot at building a style sheet mechanism for the web. Due to advancement of technology, changes have been made to CSS1 to make it more robust and this enters the CSS2 [24].

In general, CSS1 includes support for [24]

- i. Specifying font type, size, color, and effects
- ii. Setting background color and images
- iii. Controlling many aspects of text layout, including alignment and spacing
- iv. Setting margins and borders
- v. Controlling list display

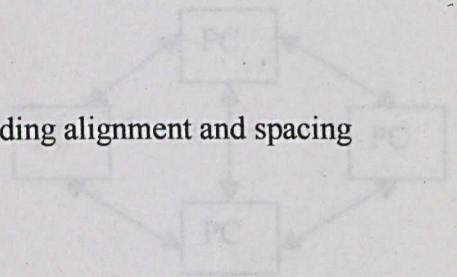


Figure 3.3 Client-server model and Peer-to-peer model architecture

CSS2 improves the implementation of many CSS1's components and expanded on CSS1. CSS2 supports include [24]

- i. Defining aural style sheet for text-to-speech browsers
- ii. Defining table layout and display
- iii. Automatically generating content for counters, footers and other standard page elements
- iv. Controlling cursor display

Because of CSS1 is more effectively supported by many web browsers, therefore it is more practical to use in the real-world development at this moment. [24]

3.5.4 Client-Server Model

In this Client-server model, two computers work together to perform a task, which a client request some needed information from a server. The server returns this information and the client acts on it.

Another type of architecture is the peer-to-peer architecture where all the computers connected has equivalent responsibilities. This architecture is not used in the system because the system needs a server to host the web pages and also to integrate the database together. Figure 3.3 shows the different architecture of Client-server model and Peer-to-peer model.

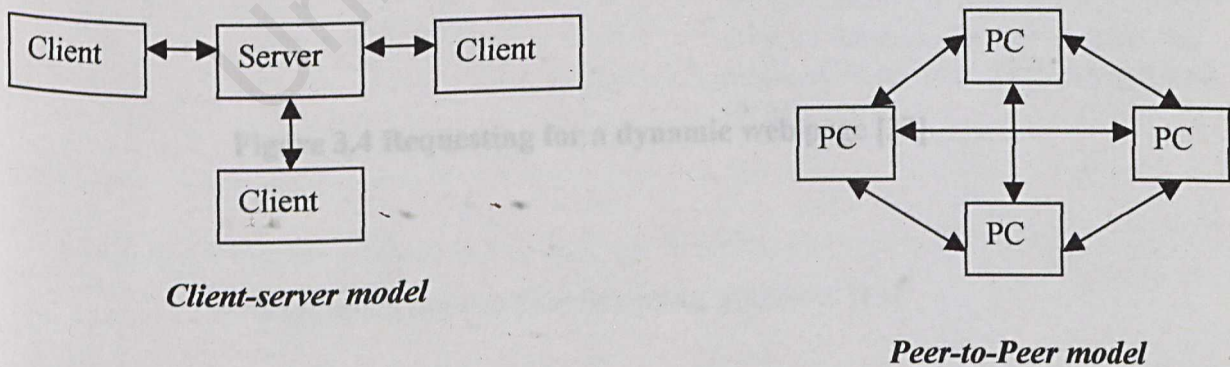


Figure3.3 Client-server model and Peer-to-peer model architecture

The Internet runs on a client-server model as well, with the Internet, the server is a particular Web Server.

The client on the Internet is the web browser and this is what happens when a static web page is visited [23]:

- i. The web browser locates the web server specified by the first part of the URL.
- ii. The client request for static web page specified by the second part of the URL.
- iii. The web server send the particular file in HTML format
- iv. The client received the HTML file and is rendered to be display on the client side.

After the request, web server sits around idle waiting for a client to request. To allow dynamic web pages to be requested, the web server must play a more active responsibility. Figure 3.4 shows the process of requesting a dynamic web page by a client [23].

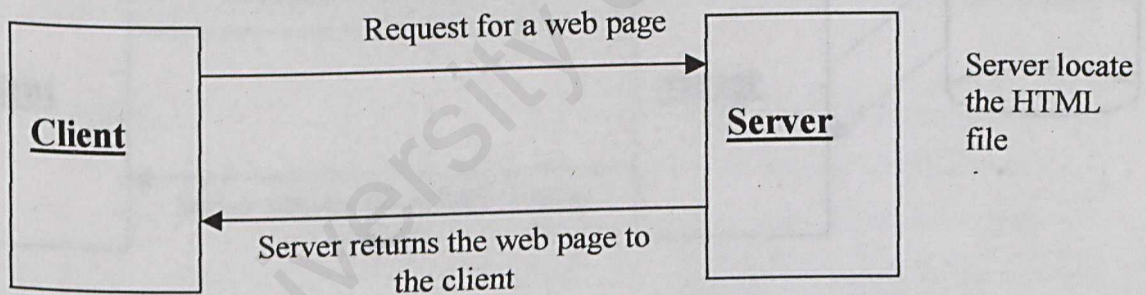


Figure 3.4 Requesting for a dynamic web page [23]

Like what was mention earlier, ASP web pages are dynamic, however, it still need the web server to process this programmatic code before sending the HTML to the client. The following steps occur when a client request for a dynamic web page [23].

- i. The client locates the Web server specified by the first part of the URL
- ii. The client then requests the ASP page specified by the second part of the URL
- iii. The web server reads the ASP file and processes the code
- iv. After the file has been completely processed by the web server, the output is sent in HTML format to the client.
- v. The client receives the HTML and can be perceive by the user.

Note that the user cannot tell the difference between an ASP page and a static web page because both receives only the HTML code. None of the programmatic code is send to the user. Figure 3.5 shows server-side scripting processes [23].

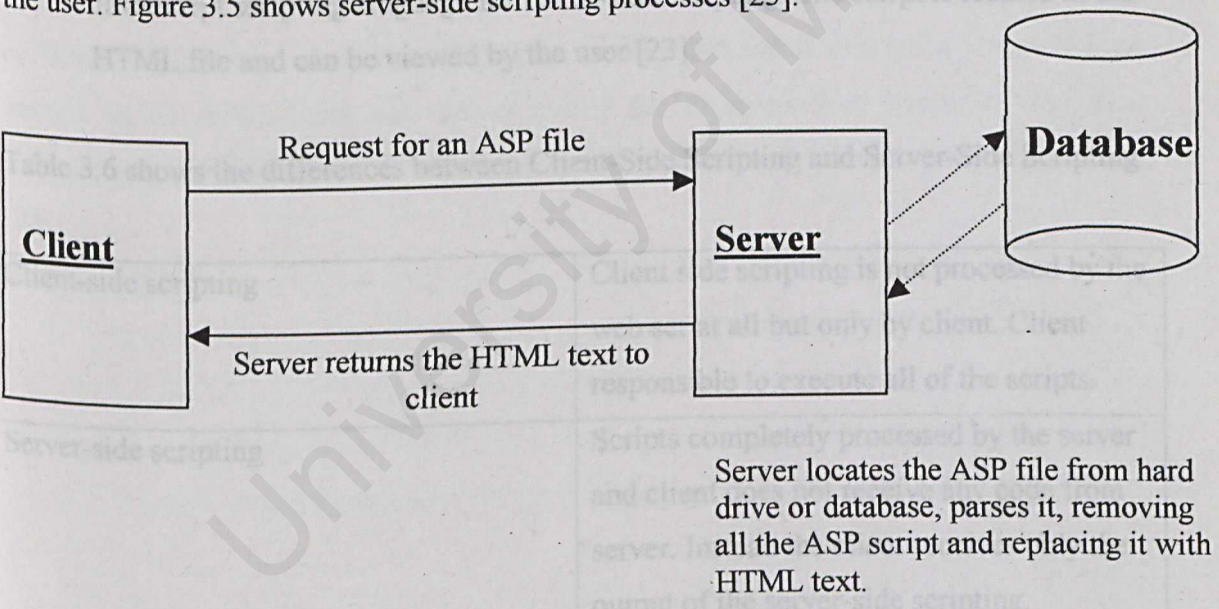


Figure 3.5 Server-Side Scripting processes [23]

3.5.4.1 Server-Side Scripting

It is important to understand that ASP is based on Server-Side Scripting and the ASP codes only exist on the server side. ASP codes will process and the result will be generated in HTML format. The client will only receive nothing much more than HTML codes. The client cannot access the ASP code.

3.5.4.2 Client-Side Scripting

Client-Side scripting is programmatic code in HTML file that runs on the browser on the client's computer. Client-Side scripting code is simply HTML code and is denoted by the <SCRIPT> HTML tag. It is commonly written using JavaScript programming language due to fact that Netscape Navigator only supports the JavaScript scripting language for client-side scripting. The script is located in the HTML file and can be viewed by the user [23].

Table 3.6 shows the differences between Client-Side Scripting and Server-Side Scripting

Client-side scripting	Client side scripting is not processed by the web ser at all but only by client. Client responsible to execute all of the scripts.
Server-side scripting	Scripts completely processed by the server and client does not receive any code from server. Instead the client receives only the output of the server-side scripting.

Table 3.6 Differences between Client-Side Scripting and Server-Side Scripting
[23].

3.5.5 Running ASP

To execute ASP pages on computers, Web server is needed. Web server processes the requested ASP page before it is sent to the client. Two things occur [23]:

- i. The pragmatic code in the ASP page is interpreted by the Web server
- ii. The web server informs the web browser that it is sending HTML information, at which point the Web server sends the output of the ASP page. The browser, receiving this raw HTML, renders it for the user.

3.5.5.1 Using Personal Web Server

When creating a professional web site, it is important that the web site run on a computer that has Windows NT or Windows 2000. But not many people run Windows NT Server or Windows 2000 on their own PC. That is why Microsoft created a stripped-down version of its professional web server called the Personal Web Server (PWS). It is intended to run on Microsoft Windows 95 or 98, or Microsoft Windows NT Workstation [23].

3.5.5.2 Using Internet Information Server

Internet Information Server (IIS) is Microsoft's Professional Web Server. The latest version of IIS is 5.0, which ships with Windows 2000. ASP 3.0 ships with IIS 5.0 and is installed automatically when IIS 5.0 is installed. [23]

If the systems work out to be running on Windows NT Server, then IIS 4.0 is needed instead of IIS 5.0. IIS 4.0 does not support ASP 3.0. So for Windows NT Server, IIS 4.0 need to be run and ASP 2.0 will be installed automatically. IIS 5.0 is shipped with Windows 2000. [23]

The benefits of IIS [23]:

- i. Integration with Windows NT Server
- ii. Comprehensive Web Server Solution
- iii. Easy-to-develop, powerful Web-based applications

3.5.6 World Wide Web

A system of Internet servers that support documents formatted in a language called HTML. Those documents may have links to other documents, graphics, audio, and video files. User can jump from one document to another simply by clicking on hot spots. Not all Internet servers are part of the World Wide Web. World Wide Web can be accessed using web browser such as Netscape Navigator and Microsoft's Internet Explorer [25].

3.5.7 FTP

FTP (File Transfer Protocol) permits user to send files to server (or receive some) and also to update (delete, rename, move, and copy) files at a server. People who create Web home pages use FTP to get their files to the server where they will be accessed. It also has a characteristic that allows file transfer services from the disk of one computer to the disk of another regardless of the operating system type [26].

Ftp is the Internet service used to transfer a data file from the disk of one computer to the disk of another regardless of the operating system type. FTP is one of the suite of protocols that are part of TCP/IP, the client/server program that every Internet server and client PC or workstation uses [26].

3.6 Summary

The methodology that is used to guide the development of the system is the Waterfall Methodology with prototyping. The method o collecting is also being defined. Requirements analysis is also being defined. There are two types of requirements defined, the functional requirements and the non-functional requirements. The types of programming language and tools that will be used are also being described. Analyzing the type of system that will be developed will help in the selection of programming languages and tools that will be used to develop the system. The systems requirements are also defined in this chapter.

4.1 Importance of System Design

System design is considered as an important part of the software development process. It is utilized and applied regardless of what kind of development model or standard being used. There are series of activities involved in system design process, namely, analyzing, coding, and testing. It is a process to ensure that it conforms to the software specifications.

Chapter 4

Information system design is defined as a series of tasks that focus on the specification of a detailed computer-based solution. It is also called the physical design [2].

4.2 Design process

System Design

Primarily, software gives a graphical profile of representation. Consequent refinement leads to a design representation that is agreeable with the source code.

The system design for the electronic document delivery system is divided into 4 parts. They are:

- i. Functional design
- ii. User interface design
- iii. Database design
- iv. Program design

4.1 Importance of System Design

System design is considered as an important part of the software development process. It is utilized and applied regardless of what kind of development model or standard being used. There are series of activities involved in system design process, namely, analyzing, coding, and testing the system or prototyping to ensure that it conforms to the software specifications and requirements, which have been defined in earlier stages.

Information system design is defined as those tasks that focus on the specification of a detailed computer-based solution. It is also called the physical design [27].

4.2 Design process

Software design is a process in which written and verbal requirements are transformed into a representation such as prototype. In other words, the process translates an abstract idea into software modules.

Primarily, software gives a condensed profile of representation. Consequent refinement leads to a design representation that is agreeable with the source code.

The system design for the electronic document delivery system is divided into 4 parts. They are:

- i. Functional design
- ii. User interface design
- iii. Database design
- iv. Program design

4.3 Functional Design

The functional design is separated into two different categories, which are;

- i. User functions
- ii. Administrator functions

The users functions and administrator function are both different. This is because the user's and administrator's requirements, view, authority and restrictions are different.

Table 4.1 and Table 4.2 show the users and administrator functions respectively.

No.	Functions	Description
1.	Login	To login in into the service
2.	Search the catalog	To search the OPAC to ensure the document is not in the library collections
3.	Links	To link to another web pages
4.	Make request	To key in request information
5.	Submit request	To submit request for material
6.	Make enquiries	To make enquiries about the service
7.	Make suggestions	To make suggestions or comments about the service
8.	Logout	To logout from the service

Table 4.1 User requirements

No.	Functions	Description
1.	View task list	To view the requests made and proceed with processing of request
2.	Search the catalog	To search the OPAC to ensure the document is not in the library collections
3.	View summary report	To view the summary of service usage
4.	Answering enquiries	To give feedback to user's enquiries
5.	Answering suggestions	To give feedback on user's suggestions and comments
6.	Notifying user	To notify user when materials arrive

Table 4.2 Administrator requirements

4.4 User Interface Design

The EDDS is a web-based system, so the user interfaces are designed according to web page style. Typically the user interface is to help users to navigate through web pages and make requests. It will also help the system administrator in doing the processing and queries. User interface provide a transparent layer between the users, administrators and the back end itself system.

Each web page will have its unique and standardize features. Web pages will be equipped with relevant links and information. The web pages will have a formal design to suites their purpose to deliver information directly and clearly. Data should be displayed in an organized manner. Furthermore, each of the interface have it's own function so arrangement of how to present these functions is also a major concern.

The EDDS interfaces are divided into two different categories, which are,

- i. Requester interface
- ii. Administrator interface

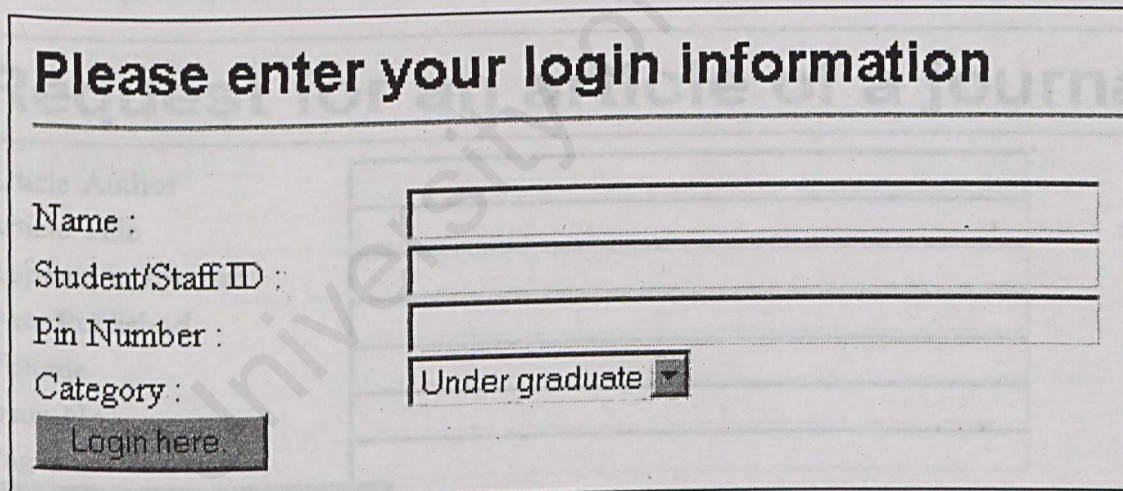
The requester interface is separated into different kinds of interfaces, each having its own information need to be delivered to the users or need to be captured by the EDDS. It is also the same criteria for the administrator.

4.4.1 Requester interface

The samples of interfaces for the requesters are depicted in the following figures:

Figure 4.1 shows the Login interface, while Figure 4.2 shows the Menu interface. Figure 4.3, 4.4 and 4.5 respectively shows the request for an article of a journal, request for a chapter of a book and making enquires interface. While Figure 4.6 and 4.7 shows the giving suggestions and logout interface.

i. Login



The login interface is a rectangular box with a title bar at the top that reads "Please enter your login information". Below the title bar, there are four input fields stacked vertically. The first field is labeled "Name :", the second "Student/Staff ID :", the third "Pin Number :", and the fourth "Category :". The "Category :" field is a dropdown menu with "Under graduate" selected. Below the input fields is a button labeled "Login here".

Name :	<input type="text"/>
Student/Staff ID :	<input type="text"/>
Pin Number :	<input type="text"/>
Category :	Under graduate <input type="button" value="v"/>
<input type="button" value="Login here"/>	

Figure 4.1 Login interface

ii. Menu

Welcome to our Electronic Document Delivery Services

Please choose one of the options available.

☐ An article of a journal

☐ A chapter of a book

Catalog Search

Proceed>>>

Figure 4.2 Menu interface

iii. Request for an article of a journal

Request for an article of a journal

Article Author	
Article Title	
Appears in Journal	
Date Published	
Volume	
Issue No.	
Pages	

Submit your request

Figure 4.3 Request for an article of a journal interface

iv. Request for a chapter of a book

Request for a chapter of a book

Chapter Author	<input type="text"/>
Chapter Title	<input type="text"/>
Title of Book	<input type="text"/>
Book Author	<input type="text"/>
Pages in Book	<input type="text"/>
Publisher	<input type="text"/>
Date	<input type="text"/>
Place Publish	<input type="text"/>
ISBN	<input type="text"/>

Submit your request

Figure 4.4 Request for a chapter of a book interface

v. Making enquiries

Enquiries

Name

:

Email Address

:

Enquiries:

Figure 4.5 Making enquiries interface

vi. Giving suggestions

Comments and Suggestions

Name :

Email Address :

Comments/Suggestions :

Figure 4.6 Giving suggestions interface

vii. Logout

Thank you for using the service.

Library Main Page

Suggestions

Make Another Request

Figure 4.7 Logout interface

4.4.2 Administrator interface

The samples of interfaces for administrator are depicted in the following figures:

Figure 4.8, 4.9 and 4.10 shows the interface to generate task list, generate summary and give feedback to user’s enquiries. While Figure 4.11 and 4.12 depicted the giving feedback to user’s suggestions and comments and making query interface.

i. Generate task list

Generate Task List

Staff Name :

Staff ID :

Duration :

From :

To :

Generate

Figure 4.8 Generate task list interface

ii. Generate report and summary

Generate Report

Staff Name :

Staff ID :

Duration :

From :

To :

Generate

Figure 4.9 Generate report and summary interface

iii. Giving feedback to user's enquiries

Enquiries Feedback

Staff Name :

Staff ID :

Enquiries :

By :

Feedback :

Submit

Figure 4.10 Giving feedback to user's enquiries interface

- iv. Giving feedback to user's suggestions and comments

Suggestions and Comments Feedback

Staff Name :

Staff ID :

Suggestions and comments :

By :

Feedback :

Submit

Figure 4.11 Giving feedback to user's suggestions and comments interface

- v. Making query

User Information Query

Staff Name :

Staff ID :

User Name :

User ID :

Submit

Figure 4.12 Making query interface

4.5 Database Design

The database for the EDDS is used to store the users information and the request information. Besides, it also stores the user’s enquiries, suggestions and comments; and also the administrator’s feedback. The database will not be use to store any materials sent by remote source. In this stage of development, only one relational database will be used instead of distributed database.

The database contains all the information captured from EDDS e-form and will be retrieved by the administrator to generate task lists, reports or summaries. Storing and retrieving data from the database will be managed by the DBMS. It provides a transparent interface between users and the database itself.

	User's information ID and Pin number
	requested
Enquiries	User's enquiries
Suggestions	User's suggestions
View task list	Task list generations
View summary	Summary generations
View enquiry	View enquiries
View suggestions	View suggestions
Make query	Retrieve user's information
Give feedback	Give feedback on user's enquiries and suggestions

Table 4.3 Different functions or sub-functions of EDDS

4.5.1 Information captured by the database and retrieved from database

For different interfaces and e-forms, there will be different data and information need to be captured. The Table 4.3 shows the different functions or sub-functions of EDDS.

Function	Type of function	Information need to be captured or retrieved
User	Login	User's information, ID and Pin number.
	Request for an article of a journal	Information about the article of journal to be requested.
	Request for a chapter of a book	Information about the chapter of a book to be requested
	Enquiries	User's enquiries
	Suggestions	User's suggestions
Administrator	View task list	Task list generations
	View summary	Summary generations
	View enquiries	View enquiries
	View suggestions	View suggestions
	Make query	Retrieve user's information
	Give feedback	Give feedback on user's enquiries and suggestions

Table 4.3 Different functions or sub-functions of EDDS

4.5.2 Tables

The data need to be stored in the database have different types and attributes. Table 4.4 shows the different data types and attributes for the user login function.

Field Names	Data Type	Size	Description
Name	Varchar	50	User name
UserID	Varchar	15	User ID
Pin	Int	10	Pin number
Category	Varchar	15	Category of user

Table 4.4 User login function data types and attributes

Table 4.5 shows the different data types and attributes for the request for an article of a journal function.

Field Names	Data Type	Size	Description
Name	Varchar	50	User name
UserID	Varchar	15	User ID
JournalTitle	Varchar	50	Journal title
ArticleAuth	Varchar	50	Article author
ArticleTitle	Varchar	50	Article title
PublishDate	Datetime	8	Date published
Vol	Varchar	5	Volume of journal
IssueNo	Varchar	5	Issue Number
ArticlePages	Varchar	10	Pages of article

Table 4.5 Request for an article o a journal function data types and attributes

Table 4.6 shows the different data types and attributes for the request for a chapter of a book function.

Field Names	Data Type	Size	Description
Name	Varchar	50	User name
UserID	Varchar	15	User ID
ChapAuth	Varchar	50	Chapter author
ChapTitle	Varchar	50	Chapter title
BookTitle	Varchar	50	Book title
BookPages	Varchar	10	Pages in book
Publisher	Varchar	50	Publisher
PublishDate	Datetime	8	Date of publish
PublishPlace	Varchar	50	Publication date
ISBN	Varchar	50	ISBN number

Table 4.6 Request for a chapter of a book function data types and attributes

Table 4.7 shows the different data types and attributes for the user enquiry function.

Field Names	Data Type	Size	Description
Name	Varchar	50	User name
EMail	Varchar	50	E-mail address
Enquiries	Varchar	500	User's enquiries

Table 4.7 User enquiry function data types and attributes

Table 4.8 shows the different data types and attributes for the user suggestion and comment function.

Field Names	Data Type	Size	Description
Name	Varchar	50	User name
EMail	Varchar	50	E-mail address
Suggestion	Varchar	500	User's suggestion and comment

Table 4.8 User suggestion and comment function data types and attributes

Table 4.9 shows the different data types and attributes for the task list generating function.

Field Names	Data Type	Size	Description
Name	Varchar	50	Administrator name
UserID	Varchar	50	Administrator ID
StartDate	Datetime	8	Starting date of task list
EndDate	Datetime	8	Ending date of task list

Table 4.9 Task list generating function data types and attributes

Table 4.10 shows the different data types and attributes for the summary generating function.

Field Names	Data Type	Size	Description
Name	Varchar	50	Administrator name
UserID	Varchar	50	Administrator ID
StartDate	Datetime	8	Starting date of summary
EndDate	Datetime	8	Ending date of summary

Table 4.10 Summary generating function data types and attributes

Table 4.11 shows the different data types and attributes for the enquiries feedback function.

Field Names	Data Type	Size	Description
Name	Varchar	50	Administrator name
UserID	Varchar	50	Administrator ID
Feedback	Varchar	500	Administrator feedback

Table 4.11 Enquiries feedback function data types and attributes

Table 4.12 shows the different data types and attributes for the suggestions and comments feedback function.

Field Names	Data Type	Size	Description
Name	Varchar	50	Administrator name
UserID	Varchar	50	Administrator ID
Feedback	Varchar	500	Administrator feedback

Table 4.12 Suggestions and comments feedback function data types and attributes

Table 4.13 shows the different data types and attributes for the information query function.

Field Names	Data Type	Size	Description
Name	Varchar	50	Administrator name
UserID	Varchar	50	Administrator ID
Name	Varchar	50	User name
UserID	Varchar	50	User ID

Table 4.13 Information query function data types and attributes

4.6 Program Design

Information system design is defined as those tasks that focus on the specification of a detailed computer-based solution. It is also called the physical design [27]. Program design emphasizes on the technical or implementation part of the EDDS [27].

4.6.1 EDDS Decomposition

A complex system is usually hard to fully understand when viewed as a whole. Therefore, in the analysis stage, we separate a system into its component subsystems, which are decomposed into smaller subsystems, until we have identified manageable subsets of the overall system. This technique is called decomposition [27].

The EDDS as whole has three identifiable subsystems. In each of these three subsystems, there are few smaller subsystems. Figure 4.13 shows the decomposition diagram for the EDDS.

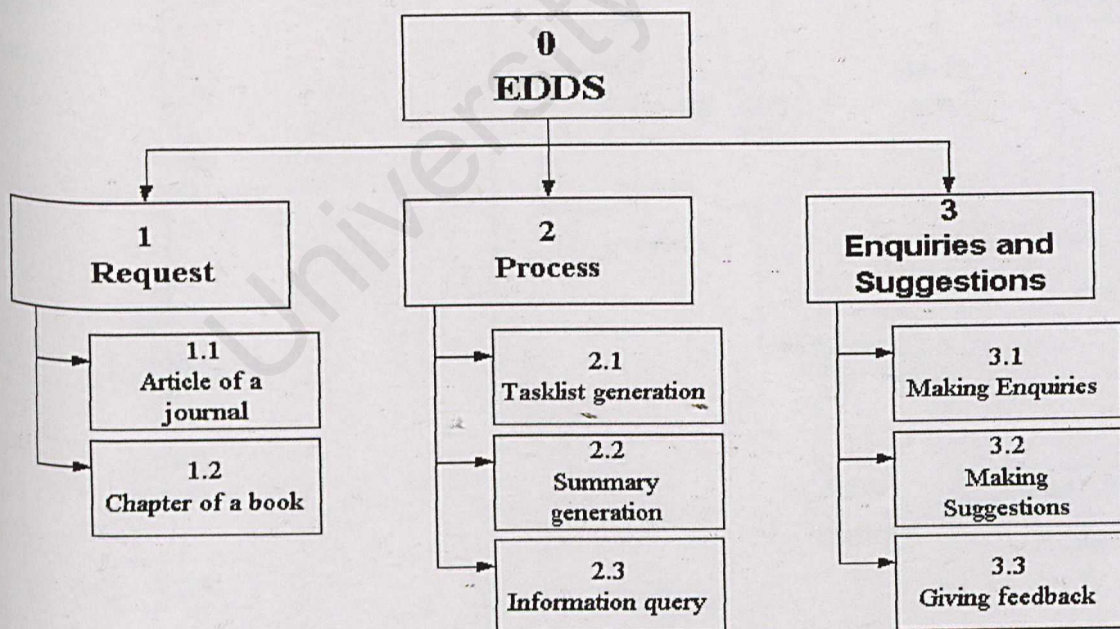
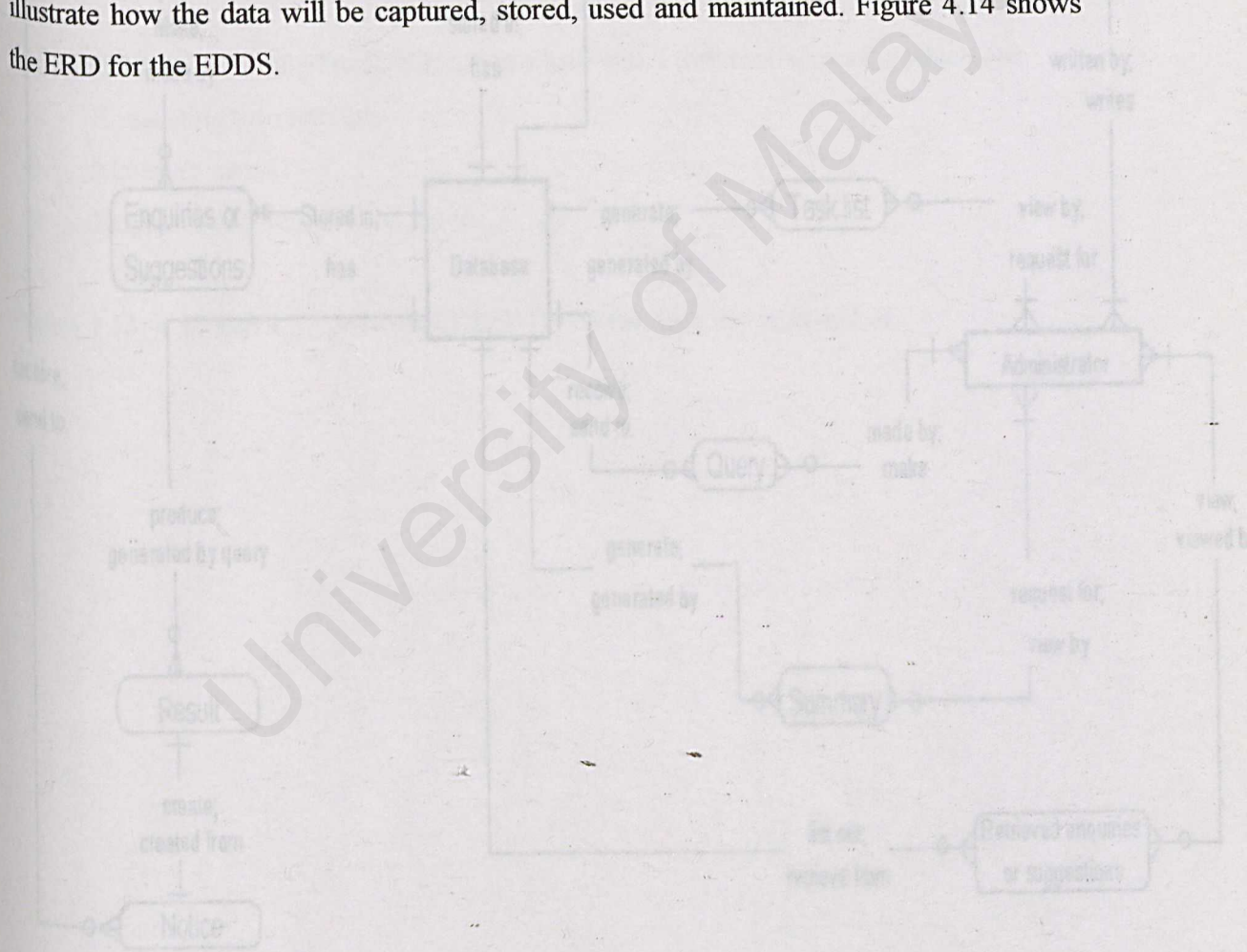


Figure 4.13 EDDS decomposition diagram

4.6.2 EDDS Entity Relationship Diagram

Data modeling is a technique for organizing and documenting a system's data. It is also sometimes called database modeling because the data model will eventually implemented as a database. The modeling process may also be recognized as information modeling [27].

The EDDS data model is represented using a graphical approach named the entity-relationship diagram or ERD. The ERD is drawn first before the data flow diagrams that illustrate how the data will be captured, stored, used and maintained. Figure 4.14 shows the ERD for the EDDS.



Entity relationship diagram

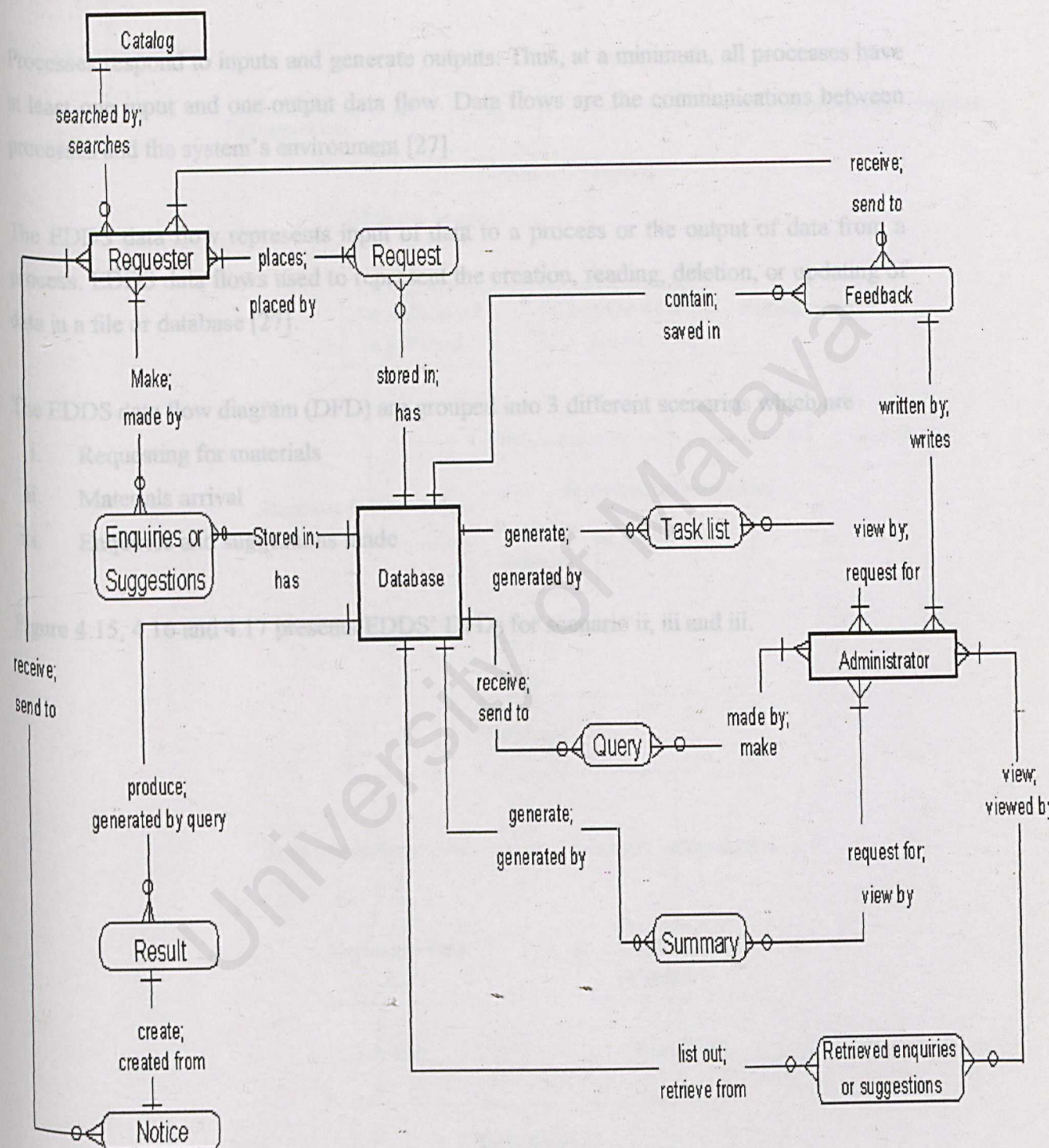


Figure 4.14 EDDS Entity Relationship Diagram

4.6.3 EDDS Data Flow

Processes respond to inputs and generate outputs. Thus, at a minimum, all processes have at least one input and one output data flow. Data flows are the communications between processes and the system’s environment [27].

The EDDS data flow represents input of data to a process or the output of data from a process. EDDS data flows used to represent the creation, reading, deletion, or updating of data in a file or database [27].

The EDDS data flow diagram (DFD) are grouped into 3 different scenarios which are

- i. Requesting for materials
- ii. Materials arrival
- iii. Enquiries and suggestions made

Figure 4.15, 4.16 and 4.17 presents EDDS’ DFDs for scenario ii, iii and iii.

Figure 4.15 DFD for EDDS scenario ii: Requesting for materials

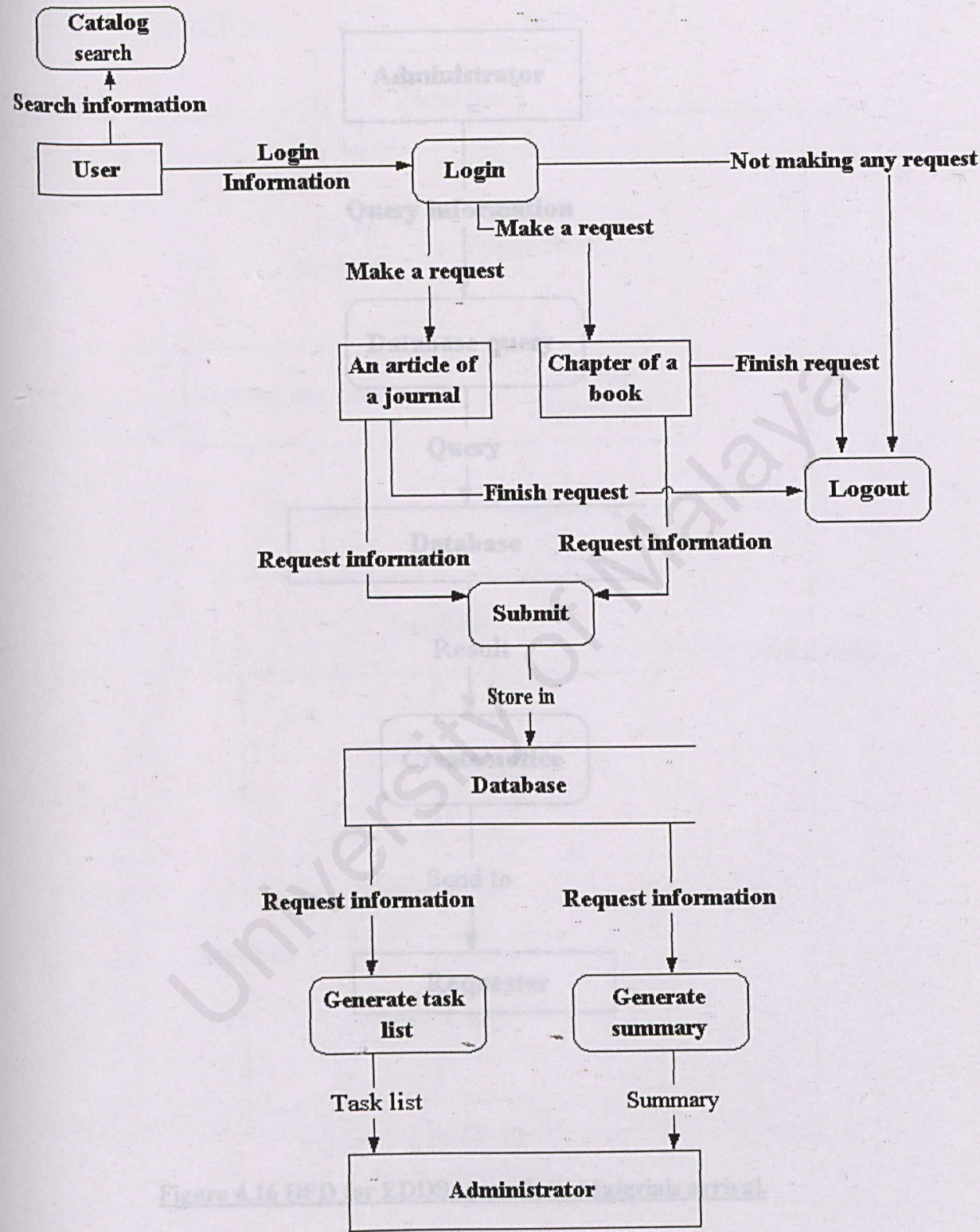


Figure 4.15 DFD for EDDS scenario i: Requesting for materials.

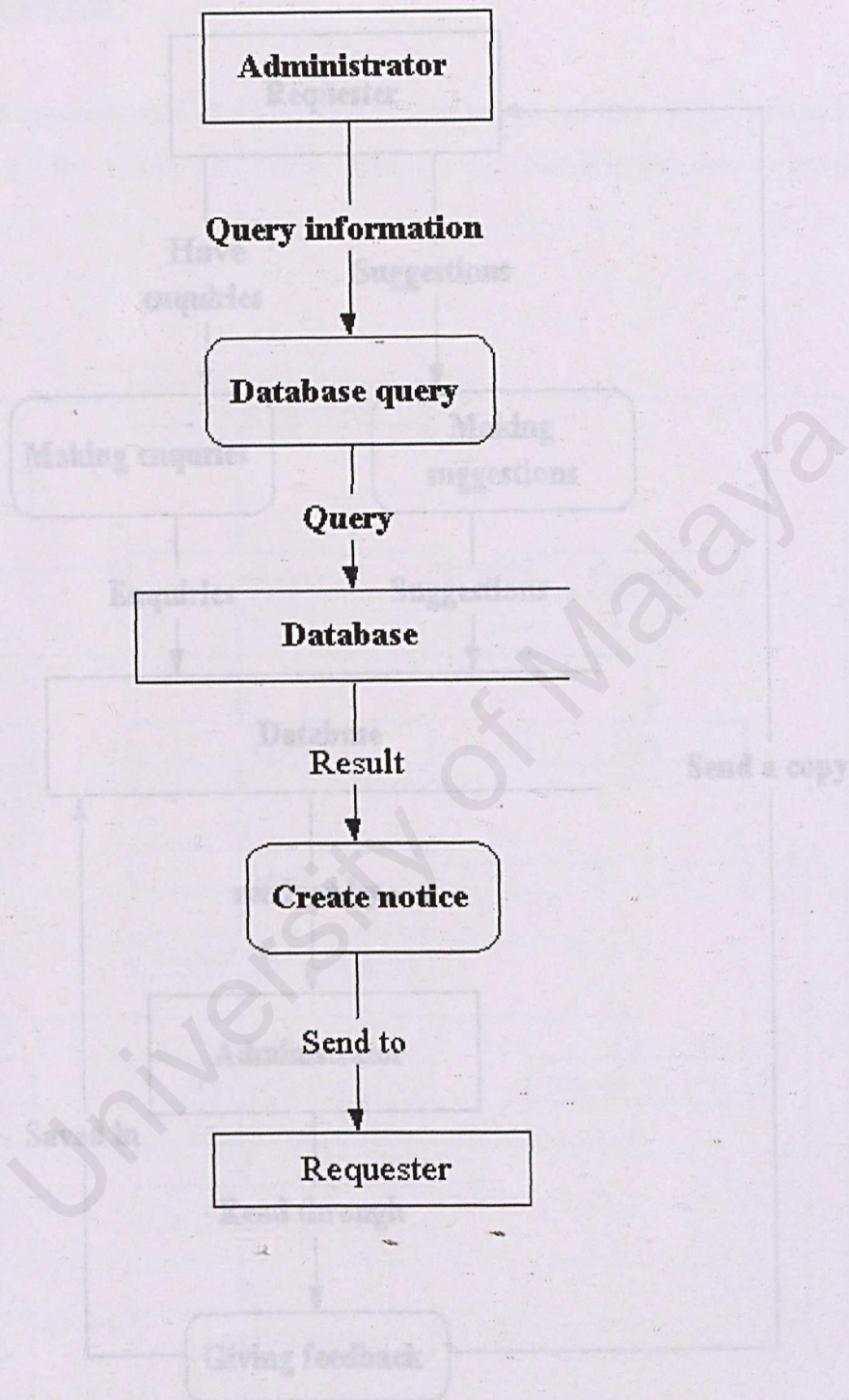


Figure 4.16 DFD for EDDS scenario ii: Materials arrival.

Figure 4.17 DFD for EDDS scenario iii: Enquiries and suggestions made

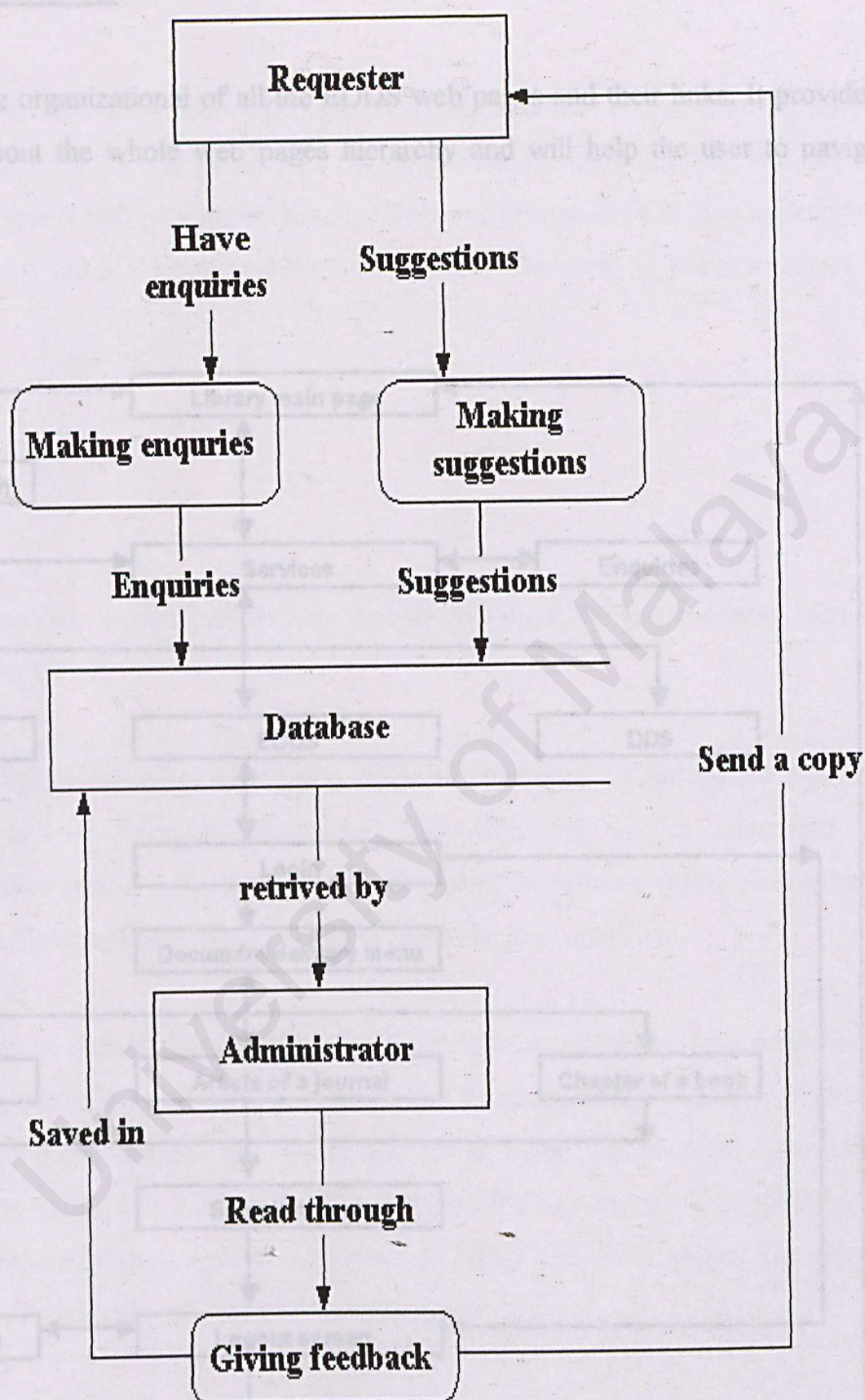


Figure 4.17 DFD for EDDS scenario iii: Enquiries and suggestions made.

4.6.4 EDDS Site Map

A site map is the organizational of all the EDDS web pages and their links. It provides a better picture about the whole web pages hierarchy and will help the user to navigate through the site.

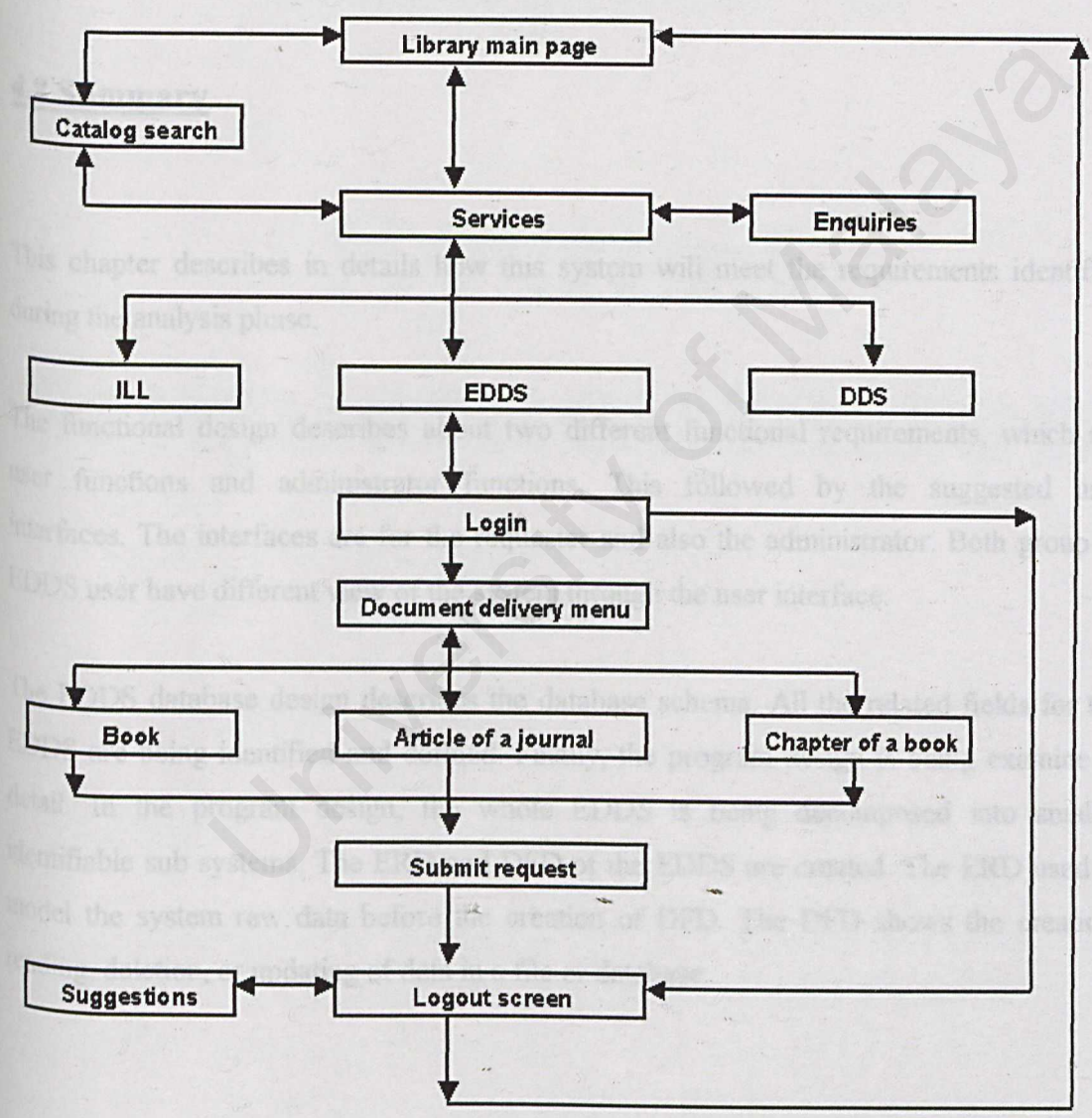


Figure 4.18 EDDS site map.

4.7 Expected outcome

It is expected that with all the functional, user interface, database and program design, the final product of the EDDS will achieve all the objectives and aim defined. Besides than that, it is hoped that it will provide all the significances promised. It is also hoped that it will introduce a new era of document delivery services in University of Malaya Library.

4.8 Summary

This chapter describes in details how this system will meet the requirements identified during the analysis phase.

The functional design describes about two different functional requirements, which are user functions and administrator functions. This followed by the suggested user interfaces. The interfaces are for the requester and also the administrator. Both group of EDDS user have different view of the system through the user interface.

The EDDS database design describes the database schema. All the related fields for the EDDS are being identified and defined. Finally, the program design is being examine in detail. In the program design, the whole EDDS is being decomposed into smaller identifiable sub systems. The ERD and DFD of the EDDS are created. The ERD used to model the system raw data before the creation of DFD. The DFD shows the creation, reading, deletion, or updating of data in a file or database.

5.1 System Implementation Process

System implementation converts the EDDS system design into program codes. Which means it focuses on implementing the solution as software [18]. There are many guidelines and standards in software coding. These guidelines not only require a programmer to write code so that it will be understandable not only to the programmer himself but to other programmers who will be working on the system design as its core reference.

Chapter 5

5.2 System Coding

There are several aspects in system coding activities. The aspects are:

System Implementation

1. Hardware Requirements
2. Software and Development Tool Requirements
3. Unit Coding
4. Module Coding
5. Database Implementation

5.2.1 Methodology

The methodology chosen for developing EDDS is the Waterfall Model with Prototypes. All the phases in waterfall model are being used as reference and guidelines. All the workflow follows the exact waterfall model, which at this stage, the EDDS system designs will be translated into a real software solution through coding.

5.1 System Implementation Process

System implementation converts the EDDS system design into program codes. Which means it focuses on implementing the solution as software [18]. There are many guidelines and standards in software coding. These guidelines not only require a programmer to write code so that it will be understandable not only to the programmer himself but to other programmers as well. System implementation uses the system design as its core reference in building the program codes.

5.2 System Coding

There are several aspects, which are considered during the system coding activities. The aspects are:

1. Methodology
2. Schedule
3. Hardware Requirements
4. Software and Development Tools Requirements
5. Unit Coding
6. Module Coding
7. Database implementation

5.2.1 Methodology

The methodology chosen for developing EDDS is the Waterfall Model with Prototyping. All the phases in waterfall model are being used as reference and guidelines. All the workflow follows the exact waterfall model, which at this stage, the EDDS system designs will be translated into a real software solution through coding.

5.2.2 Schedule

All coding activities were monitored and supervised in order to match the timeline and dateline on the EDDS development schedule. All coding activities and the testing activities, which was carried out parallel, must be planned well in order to make sure the whole EDDS development is on track. Due to time constraint, lack of expertise and limited resources, the entire unit testing has to be carried along as the system coding.

5.2.3 Hardware requirements

As stated in the EDDS system requirements, all the hardware must be prepared early and tested out so that it will deliver what the programmer intended to do during the coding activities. Error such as wrong configuration of hardware and malfunctioning hardware will drag and slow down the development and coding process.

The hardware aspects that need to be setup before the development and coding process begins are:

- i. Computer name
- ii. Computer IP Address
- iii. Internet connection
- iv. Networking Protocol

5.2.4 Software and Development Tools Requirements

All the software and development tools needed such as the Microsoft Visual Interdev, documentation software, SQL Server, Server OS and Web browser need to be installed and configure before the actual coding activities starts. The development tools must be tested out and any additional requirements, such as third party software, need to be installed before all coding activities begin.

Things that need to be setup before the coding activities begin are:

- i. SQL server
- ii. Root Directory
- iii. Relevant information folders
- iv. Other development services such as SMTP service and Microsoft service pack
- v. Graphics program
- vi. Icon program

5.2.5 Unit Coding

After hardware, software and development tools are ready, the actual coding activities begin. It started with analyzing the EDDS design and recognizing all the sub-module or sub-unit. The unit coding started off with a rough idea or image of the requirements. The prototype, which has been developed during the EDDS system design, is being refine and expands. The basic functionalities are being refine and the extra requirements which have been stated in the design or which are logically needed, are being added to the prototype.

The prototype eventually expands to its need requirements and the interface and appearance are being developed. The interface, layout and appearance depends much on the programmers view, experience and knowledge. Programmers also need to understand their target end-user needs, requirements and taste to engineer an acceptable interface. Furthermore the interface should be eye pleasing and user friendly.

For the unit coding, the basic requirements of the relevant unit are coded. The coding includes the data structure of each unit are being develop. Database connection and database data and table manipulation are also being coded. Lastly but not the least important, the unit interface is also being design and coded. All the codes are developed using the Microsoft Visual Interdev. Interdev is an automated tool to develop an ASP page. Coding time and task are made short and simple using Interdev development interface. Furthermore, Interdev provides an important feature that allow programmer to view the interface design without the need to open up a web browser. For the EDDS unit, all of them are mostly ASP pages. As we continue on to discuss EDDS unit coding, we

will use the term ASP pages to represent the EDDS unit because mostly the unit are in ASP web page format.

5.2.5.1 ASP

For the entire ASP pages developed, the entire ASP coding starts with the `<%@Language = VBScript %>` line. Some of the ASP features are listed below:

- i. Email
- ii. Session variables
- iii. Redirect

i. Email

The EDDS requires a special ASP function, which is the internal email function. This email function will only work with IIS, Windows 2000 family and Windows NT. All the email will use the SMTP service installed by those programs mentioned above. Collaborative Data Object for NT server (CDONTS), which are installed along with the IIS SMTP server, provide a convenient, easy-to-use, server-independent, and robust way to send messages from within ASP [28].

The CDONTS codes are placed in one file, used as attachment in every EDDS pages which need the email feature. The attachment is indicated by the code listed below:

```
<!-- #include file="cdonts.inc" -->
```


To set the email function, the code listed below being used:

```
Dim NewMailObj  
Dim strMailBody  
Set NewMailObj = CreateObject("CDONTS.Newmail")  
NewMailObj.From = "elec_doc_del@yahoo.com"  
NewMailObj.To = strMail  
NewMailObj.Subject = "Welcome to UM Library EDDS"  
NewMailObj.MailFormat = CdoMailFormatMime  
NewMailObj.BodyFormat = CdoBodyFormatText  
NewMailObj.Body = strMailBody  
NewMailObj.Send  
Set NewMailObj = Nothing
```

NewMailObj.From indicates the sender email address and *NewMailObj.To* indicate the receiver address. The receiver address will be taken from the database. *NewMailObj.Subject* will hold the subject title for the email.

NewMailObj.MailFormat = CdoMailFormatMime will limit the length of the message to 74 characters or fewer [28].

NewMailObj.Body line keeps the email content, which will be defined or created by the programmer. It is here where all the standardize email content being developed.

NewMailObj.Send code sends the email and the *Set NewMailObj = Nothing* will close the email function. This is needed before you can send another email.

ii. Session variables

Session variable are used to track current user information as the user moves from one page to page [28]. We need to track the user information using session variables because the web is stateless. Session variables technique is chosen because we need to maintain state for the duration of user's visits [23]. Session variable can be declared anywhere and anytime in the EDDS pages. The content can be manipulated and changed easily. The purpose of changing the information contain in each session variables is to make sure that the session variables carries the relevant information needed by the specific EDDS pages as user moves from one page to another.

Session variables are declared using the code listed below:

- i. `Session("CurrentUserID") = Request.Form("UserID")`
- ii. `Session("CurrentUserID") = objRS("UserID")`

The first code line is to populate the session variable name CurrentUserID with information capture from the form. While the second code line is to populate the session variable name CurrentUserID with information retrieved from the database.

The content of the session variables are changed using the same codes. The content will be used and the code to extract the information contain in a specific session variable are listed below:

- i. `<%= Session("CurrentUserID")%>`
- ii. `strCurrentUser = Session("CurrentUserID")`

iii. Redirect

Another ASP feature used by the EDDS pages is the redirect feature. It is useful especially we need to redirect the web pages to different web pages or the same web page with different parameter under certain condition. The redirect code is listed below:

```
If Request ("SecondTry")= "True" Then
```

```
    Response.Redirect "eddsRegister.asp"
```

```
Else Response.Redirect "eddsLogin.asp?SecondTry=True&WrongPW=True"
```

```
End If
```

If the first condition is true, the system will redirect to the eddsRegister.asp web page from the current web page. If the first condition is false, the system will redirect to the eddsLogin.asp page with query strings attach to the URL.

5.2.5.2 HTML

HTML tags are also included in the ASP pages because all the pages will be displayed online through web browser. HTML tag will format how the ASP pages will look like on the web browser. HTML tag starts with <HTML> tag.

Typically the ASP pages will have a standard HTML tags. The ASP pages will be developed with such HTML tag pairs as listed below:

```
<HTML>
```

```
    <HEAD>
```

```
    </HEAD>
```

```
    <BODY>
```

```
    </BODY>
```

```
</HTML>
```


For ASP pages with form features, HTML form `<FORM>` `</FORM>` tags are added. This will format the relevant code into a form features. Mostly the EDDS pages deals with form manipulation.

For added features and value to the form, java script is added to perform the form validation features. This is evidence when the form pages are infested with the `<Script Language = 'javaScript'>` code. The javaScript code are added in between the `<HEAD>` and the `</HEAD>` html tag. The form validation javaScript are invoked whenever a user click the submit button on the form pages. The script is a client side scripting. This means, all the forms are protected against simple incorrect or incomplete form details input by the user before they submit the form. This feature will lessen the server workload and prevent unwanted or incomplete data from getting into the system database.

Some other HTML tag used to format the form and EDDS pages interface are:

- i. `<TABLE>`
- ii. `<HR>`
- iii. ``
- iv. `<CENTER>`

The backgrounds, icons, banner, and standard attachment files are kept in the same directory, which is the root directory. All those data are organized into folders for easy reach and manipulation.

Besides HTML and ASP codes, the EDDS pages also use VBScript to manipulate data. VBScript are almost like Visual Basic codes. There are more flexible and not case sensitive if compare to other languages like Java Script. A lot of the basic VBScript features and built in functions are being used such as:

- i. Control structures such as looping and conditional logic
- ii. Formatting functions
- iii. String functions

For the database connection the codes for the connection are placed into a standard file name dbaseConn.inc. The file is attached to the EDDS pages, which need database connection and database data. The content of the dbaseConn.inc is listed below:

```
<%  
Dim objConn  
Set objConn = Server.CreateObject ("ADODB.Connection")  
objConn.Open "dsn=edds; UID=sa; Password=123"
```

```
%>
```

The UID and the Password are set during the installation and SQL configuration. For security reason, the UID and the Password can be changed.

For record set manipulation in the EDDS pages, the below code is being declare.

```
Dim objRS Set objRs = Server.CreateObject("ADODB.Recordset")
```

All the record resulting from sql query will be addressed as objRS. An example of the coding developed to make database query is listed below:

```
Dim sql  
sql = "SELECT * FROM UserInfo WHERE UserID = '" & strUserID & "'">  
objRs.Open sql, objConn, 3, 3, 1
```

Where sql is the sql statement, which will be carried out. All the database connection will be closed after all the database manipulations are being carried out with the code listed below :

```
objRs.Close  
Set objRS=Nothing  
objConn.Close  
Set objConn= Nothing
```


5.2.6 Module Coding

Each module in the EDDS consists of units that are put together. The module coding is more likely to be higher level than unit coding. Thus it is not as complex as unit coding. Mainly module coding start with making sure all units can communicate together and the units have a standard content. Each unit must be able to understand other unit if they are linked. The input from one unit to the other must have the same type and format.

For the EDDS, the inputs, which will be passed on from one unit to the other, are mostly being handle by session variables and database tables. Some extra codes needed to make sure that the units are able to communicate with each other such as sending variables through query strings from one page to another.

Next the module was organized to accomplish the module requirement. Once the module can accomplish the requirement needed, all of them will be organized into the whole EDDS system. The EDDS system contains each module logically linked and organized to accomplish the required task.

5.2.7 Database implementation

Database implementation is a process or activity that transform the database design into tables using the SQL server tools. Each table is created and designed according to the previous database table design. From time to time during EDDS coding, the database table design is changed to meet the requirements and need when arise. Database is backup after certain duration. This is to make sure that there is a backup data if the database corrupted.

5.3 Summary

The main focus of the EDDS implementation is the process of transforming the EDDS written design into a real working model. The process will need the programmer to read, understand and interpret the design so that solutions can be engineered. The EDDS implementation follows the schedule planned earlier and must be kept in the duration provided to avoid any possibilities of delay in the project schedule.

The implementation activities includes:

- i. Preparing the hardware according to requirements
- ii. Installing and configuring the software and development tools according to requirements
- iii. Unit Coding
- iv. Module Coding
- v. Database implementation

The tools used to develop and coding of the EDDS are Microsoft Interdev and SQL Server. Other elements used for coding are HTML, VBScript, ASP and JavaScript. Chapter 6 will present about the EDDS System Testing.

6.1 Importance of System Testing

System testing is conducted during and after the EDDS system implementation. The importance of system testing is listed in Table 3.4. The main purpose of system testing is to verify and validate the EDDS system. Verification is carried out to make sure all the EDDS units and modules are working properly and trouble free, while EDDS validation is to make sure all the proposed specification and functions specified in the EDDS design.

Chapter 6

System Testing

- i. Internal testing
- ii. Unit Testing
- iii. Application Testing

6.1 Importance of System Testing

System testing is conducted during and after the EDDS system implementation. The importance of system testing is listed in Table 3.4. The main purpose of system testing is to verify and validate the EDDS system. Verification is carried out to make sure all the EDDS units and modules are working properly and trouble free, while EDDS validation is to make sure all the propose specification and functions specified in the EDDS design are included.

It is said that testing is conducted during EDDS implementation because the programmer need to test the unit and module when they first coded them. This is to make sure there is no syntax error in the coding and it will do what the programmer expects it to do. The testing carried out after system implementation is to make sure that there are no logical errors in the coding. Codes might seem to be working well when there are no syntax errors, but they are yet to be sure that they will perform all tasks and function properly and as intended. When there is errors occur in the output when there are no errors in the syntax, it is said that the codes have logical error. The type of testing carried out on the EDDS system are listed below:

- i. Internal testing
- ii. Unit Testing
- iii. Application Testing

6.2 Internal testing

Internal testing as what the name suggested, is testing all the internal EDDS elements. The elements is said to be internal when they are invisible to the user. It can describe the EDDS backend where user have no idea what is happening when they input something or when they click on some button. Elements that are not visible to user includes:

- i. Database connection
- ii. Data and information passing through different EDDS pages
- iii. ASP
- iv. VBScript

Those internal elements are tested so that they will deliver and serve the purpose of what they should do.

6.2.1 Database connection testing

The database connection is tested using dummy data. For the testing the database connection purpose, the programmer separated the testing modules into several parts. The part tested are:

- i. Opening database connection
- ii. Making database query
- iii. Retrieving data from database
- iv. Updating database data
- v. Closing the database connection

6.2.1.1 Opening database connection

The database connection is tested using an indirect method. Logically, if the database connection is established, data query can be made and we can view the result by accessing the data.

EDDS database connection is controlled by the `#include file="dbaseConn.inc"` line. When there is no error about EDDS page is being viewed through the web browser, it is sure that the database connection attachment line is working and detectable.

By looking at absence of error on the web browser does not really test out the connection. Queries are being made right after there is no error prompted by the web browser about the database connection. The result of the query will be displayed on the web page. If there are no errors about the query constructed, the result is displayed on the web page. Should there be error about the SQL query, the query statement is constructed and corrected.

6.2.1.2 Making database query

Although there is no error from the database connection testing, it doesn't really test out whether the result from a query is logic and error free.

From the database design we can expect what are the results by constructing SQL query. The queries are carried out and the results are viewed on web page after further manipulation which will be discuss in the next part. There is no standard or any written statement about what are the expected outcomes from a query. Ways to test the query result are through programmer perception and running the same SQL query using the SQL function, which is the SQL pane.

If the query can be made and the result from the query can be displayed without error, then a conclusion can be made the query is correct. But it may be sometimes not correct because the data used to test the query maybe small in amount and not complex. The query should be tested out not only during system testing but also after system acceptance and delivery.

6.2.1.3 Retrieving data from database

This testing is different from query testing. The main purpose of this testing module is to evaluate that the data could be taken out from the database after they are kept and stored in the database and stored in a variables for further processing and manipulation. This part is more to testing the information retrieval from the database. It is different from query in the sense that query only get back it's result in a records format without storing it in variables.

6.2.1.4 Updating database data

Data updating testing is carried out when there are command lines involved in changing or appending existing data in the database. The way to test whether the updating process is correct is looking at the database data before and after updating process. The programmer have make note about what the data in the database look like before the updating process. Soon after that, execute the update command and look back at the database data. The programmer will check whether the updating is correct, as what it suppose to be. Updating the EDDS database is carried out to change the request stage and updating user profile and password.

6.2.1.5 Closing the database connection

After the entire database manipulation complete, the connection must be closed to avoid any further unauthorized database manipulation. This is a way to make sure that the database is secure.

6.2.2 Database testing

In the EDDS system, database manipulation is carried in most of the functions and modules. One of the modules is the updating user profile module.

The updating user profile module codes are listed below:

```
<%@ Language=VBScript %>
<!--#include file="dbaseConn.inc"-->

<%
    Dim UID
    UID=Session("CurrentUserID")
    Dim sql
    sql = "SELECT * FROM UserInfo WHERE UserID =" & UID & ""
    Dim objRS
    Set objRs = Server.CreateObject("ADODB.Recordset")
    objRs.Open sql, objConn, 3, 3, 1
    Dim oldPassword
    Dim checkPassword
    checkPassword=Request.Form ("Password")
    oldPassword = objRs("Password")
    if (checkPassword<>oldPassword) then

        Session("returnCurrentUserID") = Request.Form("UserID")
        Session("returnCurrentUserName")=Request.Form ("Name")
        Session("returnCurrentUserSemAdd") = Request.Form("SemAdd")
        Session("returnCurrentUserMailAdd") = Request.Form("MailAdd")
        Session("returnCurrentUserContact") = Request.Form("Contact")
        Session("returnCurrentUserEmail") = Request.Form("eMail")
        Response.Redirect "UpdateProfile.asp?Password=invalid"
```


else

'Update relevant records in UserInfo table

```
objRs("UserID") = Request.Form ("UserID")
objRs("Name") = Request.Form ("Name")
objRs("Password") = Session("CurrentUserPassword")
objRs("SemAdd") = Request.Form ("SemAdd")
objRs("MailAdd") = Request.Form ("MailAdd")
objRs("Contact") = Request.Form ("Contact")
objRs("eMail") = Request.Form ("eMail")
objRS.Update
```

```
Session("returnCurrentUserID") = ""
Session("returnCurrentUserName")=""
Session("returnCurrentUserSemAdd") = ""
Session("returnCurrentUserMailAdd") = ""
Session("returnCurrentUserContact") = ""
Session("returnCurrentUserEmail") = ""
```

'Update relevant records in journal table

```
sql = "SELECT * FROM Journal WHERE UserID ="" &UID & ""
Set objRs = Server.CreateObject("ADODB.Recordset")
objRs.Open sql, objConn, 3, 3, 1
```

While not objRS.EOF

```
objRs("UserID") = Request.Form ("UserID")
objRs("Name") = Request.Form ("Name")
objRs("eMail") = Request.Form ("eMail")
objRS.Update
objRS.movenext
Wend
```

'Update relevant records in bookchap table

```
sql = "SELECT * FROM BookChap WHERE UserID ="" &UID & ""
Set objRs = Server.CreateObject("ADODB.Recordset")
objRs.Open sql, objConn, 3, 3, 1
```

While not objRS.EOF


```
objRs("UserID") = Request.Form ("UserID")
objRs("Name") = Request.Form ("Name")
objRs("eMail") = Request.Form ("eMail")
objRS.Update
objRS.movenext
Wend
```

```
'-----
'Update      relevant records in feedback table
'-----
```

```
sql = "SELECT * FROM Feedback WHERE UserID ='" & UID & "'"
```

```
Set objRs = Server.CreateObject("ADODB.Recordset")
objRs.Open sql, objConn, 3, 3, 1
```

```
While not objRS.EOF
objRs("Name") = Request.Form ("Name")
objRs("eMail") = Request.Form ("eMail")
objRS.Update
objRS.movenext
Wend
```

```
'-----
'Reset the session variables
'-----
```

```
Session("CurrentUserID") = Request.Form("UserID")
Session("CurrentUserName")=Request.Form ("Name")
Session("CurrentUserPassword") = Request.Form("Password")
Session("CurrentUserSemAdd") = Request.Form("SemAdd")
Session("CurrentUserMailAdd") = Request.Form("MailAdd")
Session("CurrentUserContact") = Request.Form("Contact")
Session("CurrentUserereMail") = Request.Form("eMail")
```

```
objRs.Close
Set objRS=Nothing
```

```
objConn.Close
Set objConn= Nothing
Response.Redirect "UserProfile.asp"
End If
%>
```

```
<HTML>
<BODY>
</BODY>
</HTML>
```


All database testing is carried out at the same time to be able to recognize any errors.

Database connection is being made by the code below:

```
<!--#include file="dbaseConn.inc"-->
```

The query is being formulated in the line below:

Dim sql

```
sql = "SELECT * FROM UserInfo WHERE UserID ='" &UID & "'"
```

Retrieving data from database is carried out by the line below:

```
oldPassword = objRs("Password")
```

Updating a database records or data is carried out the the lines below:

```
While not objRS.EOF  
objRs("UserID") = Request.Form ("UserID")  
objRs("Name") = Request.Form ("Name")  
objRs("eMail") = Request.Form ("eMail")  
objRS.Update  
objRS.movenext  
Wend
```

objRS.Update command will execute all the updating function.

Closing the database connection is carried out by the code below:

```
objRs.Close  
Set objRS=Nothing  
objConn.Close  
Set objConn= Nothing
```

Running the whole module and evaluating the result will carry out the entire database testing. If the module produces the intended result, then database manipulation coding are

error-free. The same approach is used to test out other modules that are involved database manipulation.

6.2.3 Data and information passing through different EDDS pages

The data are passed through EDDS pages using session variables. Session variables are tested their integrity by using an indirect method. The content of each session variables are invisible to the user and also programmer.

The way to look at the content of the session variable is by copying or assigning their content to a variable and then display out the content through the web page. The same procedure is repeated to the same session variables in different pages where they exist. If the content changes without any authorize manipulation, then an error occurs. To rectify this problem, the programmer traced back the session variables from where the error is notify up to the session variable declaration and assignment.

Logically, each session variables are unique if they have different name and the content should be the same unless there is any changes that apply to it.

The command line to test the category of user showed how the testing is carried out.

Each administrator page has a command line that will differentiate who is viewing the page by looking at the User ID. The User ID is stored in the session variable listed as

```
Session ("CurrentUserID").  
<% Dim strCurrentUser  
strCurrentUser = Session("CurrentUserID")  
  
If strCurrentUser <> "admin" then  
%>  
<P align=center>  
<Font size=5 color=red>  
<B>You are restricted to view this page!</B></FONT></P>  
  
<% Else %>
```


`<BODY Text = "black" background = "Pic\login_back.GIF">`

The content of the session variables *Session ("CurrentUserID")* is not visible, so therefore a value need to be stored the value in a variable using the code *strCurrentUser = Session ("CurrentUserID")* first.

The codes test whether the current user is an authorized person to view the content. If the user is not an administrator, the whole content or body of the page will not be displayed.

This evaluation part is carried out by the code listed below:

```
If strCurrentUser <> "admin" then
```

The whole testing is carried by login in as administrator and tries to view the entire administrator permitted page. Log out from administrator status after that. Login again as normal user and try to view all the pages again. If there is restriction to view those pages, that means the content of the session variable is correct and error free.

The logout function will destroy all the session variables using the code listed below:

```
<% Session.Abandon() %>
```

It is very important that all users must logout from the system after using other people profile. The entire session variable used after the logout will have new content.

6.2.4 ASP codes

The purpose of testing the ASP codes is to make sure that the ASP codes perform correctly. ASP codes can easily recognized with the `<% %>` delimiters. Mostly the code `Response.Write()` is tested out. It is tested and evaluated by looking at the result through web page. If the code could display the intended output in the web page then it is error free. Sometimes the code is used not for output in the web page but as input for other functions. Looking at the final result from the function will test this type of code

out. If the result is correct then the code is error free. Besides than testing the ASP codes contain in the delimiters, the numbers of pairs of delimiters also need to be counted. If the number of delimiter is not even, error will occur.

are listed below:

i. Loop

ii. If Then, Else

The numbers of loop need to be tested so that it will produce the intended result. If the loop is out of bound, it will result an error. Loop testing is carried out on the graph generation function in the generate summary module which is listed below:

```
for i = 0 to 3
```

```
Response.Write("<TD VALIGN=BOTTOM ALIGN=CENTER WIDTH=" & widthpercent & "% >" & labels(i))
```

```
Response.Write("</TD>")
```

```
Response.Write("<TD VALIGN=BOTTOM ALIGN=CENTER>")
```

```
Response.Write("</TD>" & chr(13))
```

```
next
```

The loop will write out 9 labels on the x-axis of the graph. If there is error in the number of labels on the graph, an error has occurred in the loop. The number of labels can be viewed through web browser.

6.3 Unit Testing

Unit testing is carried out to make sure that all user visible component work well [22].

Here we consider all user visible components as what the user can see through web browser and also the source code that could be seen when they click the option view source on the web browser. The components that are visible to user are listed below:

i. HTML code

ii. JavaScript

6.2.5 VBScript

Mostly the VBScript tested is the control structures. Some of the control structures used are listed below:

- i. Loop
- ii. If Then, Else

The numbers of loop need to be tested so that it will produce the intended result. If the loop is out of bound, it will result an error. Loop testing is carried out on the graph generation function in the generate summary module which is listed below:

```
for i = 0 to 8
```

```
    Response.Write("<TD VALIGN=BOTTOM ALIGN=CENTER WIDTH= " & widthpercent+1.0& "% ">" & labels(i))
    Response.Write("</TD>")
    Response.Write("<TD VALIGN=BOTTOM ALIGN=CENTER>")
    Response.Write("</TD>" & chr(13))
```

```
next
```

The loop will write out 9 labels on the x-axis of the graph. If there is error in the number of labels on the graph, an error has occurred in the loop. The number of labels can be viewed through web browser.

6.3 Unit Testing

Unit testing is carried out to make sure that all user visible component work well [22]. Here we consider all user visible components as what the user can see through web browser and also the source code that could be seen when they click the option view source on the web browser. The components that are visible to user are listed below;

- i. HTML code
- ii. JavaScript

iii. Links

6.3.1 HTML Code

HTML codes are directly linked with the user interface. The testing for HTML indirectly will test out the user interface. HTML tags control user interfaces. User interface need to be tested so that it will display the things that it need to display on the web page. The EDDS form is an example of HTML coding. The form must be tested out so that it will include all the fields needed for it to serve its purpose. If there is an error in display, the HTML codes need to be rectified.

6.3.2 JavaScript

JavaScript is considered visible to user because we can view the script through a web browser if compare to ASP codes. JavaScript used in EDDS system are mainly for form validation. Form validation codes for the creating a notice module are listed below:

```
<script language="JavaScript">
function validateForm() {
var Reference=document.NoticeForm.reference.value

if(Reference=="")
{
    alert("Please enter the reference number.")
    document.NoticeForm.reference.focus()
    return false
}
}</script>
```

This portion is visible to user if they view the source code through their web browser. The script can be recognize by their script tag,
<script language="JavaScript">....</script>.

The script is tested out by purposely leaving the reference field empty. As soon as the user click the submit button, the script is invoked. It will notify the user if the reference

field is empty. If there is no notice about the field is empty when the user click the submit button, then there is an error in the script.

6.3.3 Links

Moving around the web pages according to their logical hierarchy will test the links. An error can be detected if the logical flows of web pages derail from what have been proposed. The programmer then checked the link statement to make sure that it contains an existing web page as its destination and also the correct page. The EDDS comprised a logical flow of web pages in the User registration module. The flow of the module listed in figure 6.1:

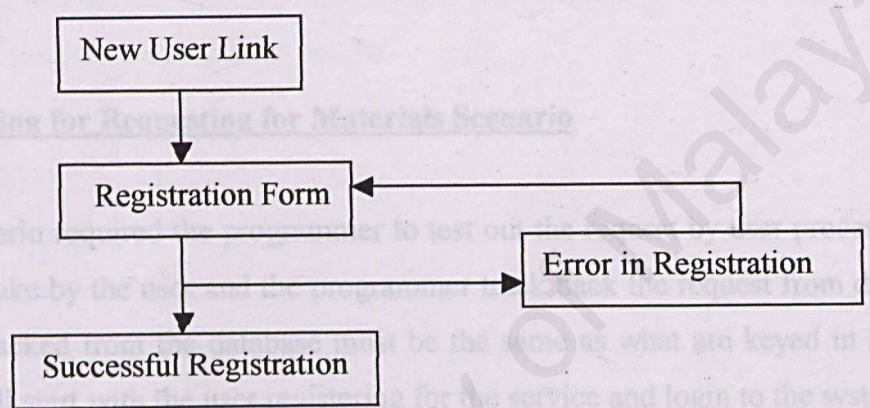


Figure 6.1 Logical Web Pages Flow for Registration Module

The links are tested out by two different cycles. The first cycle will be making a successful registration. If the links flow as what is stated in figure 6.1 for a successful registration, then there is no error in the links. The second cycle will be making an unsuccessful registration. If the links flow as what is stated in figure 6.1 for an error in registration, then there is no error in the links either. If there is a derail from the logical flow for both cycles, the links statement must be revised.

6.4 Application Testing

Application testing is carried out to make sure the EDDS system can complete all the 3 different scenarios listed in the EDDS system design. The 3 scenarios are:

- i. Requesting for materials
- ii. Materials arrival
- iii. Enquiries and suggestions made

All the 3 scenarios are tested with dummy data to trace any error. After all scenario being tested and found error free, it will be tested with real data that could be found from the Internet.

6.4.1 Testing for Requesting for Materials Scenario

This scenario required the programmer to test out the request by user process. A request will be made by the user and the programmer track back the request from database. The request tracked from the database must be the same as what are keyed in by user. The testing will start with the user registering for the service and login to the system using the registered User ID and Password. Then user will proceed with making a request for an article of a journal. The programmer retrieved back the entire request made by the user to check the request is the same as what the user want. The request must be able to manifest itself in the task list generated and summary as a request received. If any error occurs during the process, it will be tracked back to its module and unit to find the error source.

6.4.2 Testing for Materials Arrival Scenario

Using a dummy notice that says that the material has arrived from the source tests the materials arrival scenario. The administrator should be able to make query to the database and retrieve back relevant record for that particular request. And creating a notice, either notice or a pickup notice to notify the requester, will continue the testing. The requester should be able to receive the notice. If there is no error occur in this testing, means that

the EDDS has no error in this testing. Should there be any error found, it will be tracked back to its module and unit to find the error source.

6.4.3 Testing for Enquiries and suggestions made Scenario

Using a dummy feedback that is submitted to the EDDS will start the testing for the enquiries and suggestions made Scenario. The administrator should be able to make query to the database and retrieve back the feedback. The test continues by replying the feedback and check whether the user receives the reply and a copy of the reply is being saved in the database. If there is no error occur in this testing, means that the EDDS has no error in this testing. Should there be any error found, it will be tracked back to its module and unit to find the error source.

6.5 Summary

The types of testing carried out are:

- iv. Internal testing
- v. Unit Testing
- vi. Application Testing

Testing is carried out during the EDDS implementation stage and also the testing stage. Both stages testing focused on different part. During the EDDS implementation stage, internal testing and unit testing are carried out mostly. During EDDS testing stage, application testing is the main concern.

The purpose of testing is to detect any error and rectify it before the system delivery. If there is any critical error found during system delivery, it will be rather difficult to rectify them because it need extra time to carry out the error correction. This will cause delay in system delivery.

The testing stage verified and validates all the EDDS functions, which are being stated in the EDDS design documentation. The final chapter 7 presents EDDS system evaluation.

7.1 Importance of System Evaluation

There are several errors and problems encountered during the EDDS development stages. Understanding the problems and doing research with the help of some people and Internet solve these errors and problems. Besides that, a lot of analysis about web-based system development has been carried out in order to have better understanding of the concept of web-based system. Hence, the EDDS strengths, limitation and future enhancements were identified.

Chapter 7

7.2 Problems Encountered and Solution

There are several problems encountered throughout the EDDS development stages. Most of the problems encountered were solved in order to make sure that EDDS will function properly and meet the requirements. The problems and solutions are listed below.

System Evaluation

7.2.1 Development Tools Selection

There are a lot of web-based system development tools available and considered to develop the EDDS. Problems arise in selecting the suitable development tools for the EDDS because each tool available has their own capabilities and constraints. It also depends a lot on what type of web-based system need to be developed such as a e-commerce system, web services or just a normal web-based system which offers information.

Several ways have been taken to solve this problem including seeking advice and opinion from the project adviser, friends and course mate which are also developing the similar system. In addition, research and studies about the various tools available also been done before making the final selection of the tool needed to develop the EDDS.

7.1 Importance of System Evaluation

There are several errors and problems encountered during the EDDS development stages. Understanding the problems and doing research with the help of some people and Internet solve these errors and problems. Besides than that, a lot of analysis about web-based system development has been carried out in order to have better understanding of the concept of web-based system. Hence, the EDDS strengths, limitation and future enhancements were identified.

7.2 Problems Encountered and Solutions

There are several problems encountered throughout the EDDS development stages. Most of the problems encountered were solved in order to make sure that EDDS will function properly and meet it's requirements. Some of the problems encountered are listed below.

7.2.1 Development Tools Selection

There are a lot of web-based system development tools available and could be used to develop the EDDS. Problems arise in selecting the suitable development tools for the EDDS because each tool available has their own capabilities and constraint. It also depends a lot on what type of web-based system need to be developed such as a e-commerce system, web-services or just a normal web-based system which offers information.

Several ways have been taken to solve this problem including seeking advice and opinion from the project advisor, friends and course mate which are also developing the similar system. In addition, research and studies about the various tools available also been done before making the final selection of the tool needed to develop the EDDS.

7.2.2 Lack of Knowledge in ASP

ASP programming especially in developing web-based system are not taught in our course syllabus. Therefore, the understanding of web-based system may show some misunderstanding.

The most crucial problems arise from the lack of knowledge was irregularities and unorganized coding style and methods. There is no standard style been defined and used. The reasons for this problem are difficulties in understanding different example codes and dissecting example codes available in reference books.

The ways to overcome these problems are seeking friend's help and trying to understand the coding using longer time. Besides than the Internet also offer a lot of resources about ASP coding and style. Eventually by browsing through related sites, part of the solutions for this problems were found.

7.2.3 Lack of SQL Basic

The database course taught in our syllabus stressed on database concept rather than SQL statement and formulation. This aspect has been a problem in the EDDS development because the database used is SQL server. The reason of choosing the SQL server is because it is more powerful and scalable than other small-scale database program such as MS Access.

There are a lot of resources available in the Internet about SQL. Most of the SQL command could be found explained in the Internet and this has become one of the solutions for this particular problem encountered. Besides than that, advice and opinion from friends also helps to overcome this problem.

7.2.4 Lack of ASP Resources in University of Malaya

ASP is a well known in web-based system development. But the library of University Malaya does not have ample of up to date resources for ASP reference books. User is borrowing most of the relevant and up to date books and the booking queue is very long. Most of the users do not have enough time to wait for the books to be returned by the borrower.

Instead of waiting for the books, we turned on to electronic reference. Electronic resources could be obtain through the Internet and it is much more convenient and up to date. All the final year project labs are equipped with Internet connections, so this makes the problems easier to solve.

7.2.5 Lack of Hardware and Software Configuration Knowledge

The hardware and software needed to develop the EDDS need to be configured in order to meet its development requirements. For example, the database connection needs to be configured before it can be used. These problems have been overcome with help from course mates who is using similar hardware and software settings. Besides that, help files and reference from the Internet also being used to solve the problem.

7.3 Evaluation by End Users

Due to time constraint, only few people from the faculty of computer science and information technology and other faculty tested the EDDS. They tested out the system and comments and suggestions have been suggested to improve the EDDS design. Mostly the end user commented about the interface color scheme and layout. They suggested that the interface should be more colorful rather than monotone. Changes have been made to the interface color scheme and layout by adding in graphical background and banner in each page. The layout of the icons also has been commented. They requested that the icons are arranged in tabular format rather than a list.

Besides the layout and interface, some of the user also suggested some extra features such as edit user profile features and change password features. All the suggestions, which could be implemented, have been done and which could not be implemented is suggested as future enhancement.

7.4 System Strengths

The EDDS strengths were recognized and identified. The EDDS strengths are described below.

7.4.1 User-friendly Interface

The EDDS have simple user interface which could be understand and used by new user. There are many features included to make the interface user-friendly such as text box, buttons, links and icons. These features will help the user to navigate and used the EDDS. User-friendly interface will help to reduce the total cost of developing and implementation of the EDDS because we can reduce the training cost and training time needed.

Most of the EDDS web pages are equipped with links and icons to help the user inputting information into the system and lessen user's effort in navigating through the system.

7.4.2 Easy to Use and Easy to Learn

The user-friendly interface will ease the user effort in understanding and using the system. Furthermore, the processes of the EDDS are simple and most users have the idea of the whole process if they used the manual document delivery services. The EDDS is actually an automated manual document delivery services.

7.4.3 Provide Database Access

All data needed by the EDDS are captured using electronic forms. All the data collected are stored and organized by SQL server. It provides a real-time database management. Changes made can be updated as soon as the changes being submitted. Besides, query for data from database is also fast.

7.4.4 Simpler Than Manual System

As what have been mentioned just now, the EDDS is an automated version of the manual document delivery system. But the EDDS is much more simpler to use and maintain compare to the manual system. EDDS provides a paperless environment and one point data-entry. Data entered by user and will be duplicated or retrieved whenever they are needed instead of requiring the user to re-type in the same information.

7.4.5 Less Error

Due to the one point data entry, the probability of error occurring in the request will be lessened. Hence, the effort and manpower needed to process a request will be less and the time need to process a request is reduced due to less error.

7.4.6 System Transparency

The EDDS is transparent to the user. The user will have no idea how their request are being handle and processed. In fact, the user does not need to know the underlying structure of the system and database in making their request. This is important so that the EDDS will not confuse the user.

7.5 System constraints

Although much effort has been made to build the EDDS to meet its requirements and minimizing the errors, the EDDS still have it's own constraints. These constraints could not be avoided due to lack of experience in web-based system development and time limit. The EDDS constraints are listed below.

7.5.1 Lack of Advance Security Features

The security features implemented in the current system are the most basic security features involving the session variables manipulation. Sessions variable are used to determine whether a user is granted to have access to the administrator functions or not. There are no advance security features implemented because of time constraint. In fact for a web-based system, security issue is one of the most important issue in web based system development.

7.5.2 Not a Full Working System, More to a Prototype

The EDDS, which has been developed, is more towards a prototype rather than a complete system. The EDDS although is a prototype, it has all the main functions and requirements. Enhancements need to be made if the EDDS would like to be implemented as real time service. From time to time, there are requests from the system adviser, system moderator and also system testing respondents to add in other features and functions. But due to time constraint, some of these additional requirements cannot be implemented.

7.5.3 No Date and Time Line Validation

The EDDS do involved in capturing time and date input by the user. There is no date and time validation code implemented in the EDDS because lack have ASP and VBScript mastery and time constraint. The date format input by the user must be in MM/DD/YY

format where else, our environment and society are used to DD/MM/YY format. This will be a problem to the user if they do not read or follow the guidance given in the manual or interface when they type in a date especially.

Besides than date format, in order to generate summaries and task list, the administrator need to also make sure the time line inputted is logic. The system is lacked of complex coding, which can check whether the time line is logic. If the user input the wrong time line or illogic time line, the outputs generated would not be correct.

7.5.4 No Other Language Support

The EDDS will use English as its main communicating medium with the users. The reason of using English is the problem faced in translating the English terms into Bahasa Malaysia.

7.5.5 No Complex Error Detection

The error detection features are implemented in the EDDS form. Forms validation codes are being used to make sure that no error found in user input. But the form validation codes are very basic and it could not detect illogic input and format of user input.

7.5.6 Email Attachment Using ASP

The EDDS sending notice functions uses the Outlook Express as its main email program because of problems implementing ASP codes in doing email attachment. There is code for doing email attachment but it is very hard to implement due to lack of mastery in ASP and VBScript. There is another solution found through research done, but the solution is not so advisable, which is using third party emailing software to be incorporated in the EDDS. The approach could be expensive because the third party emailing software need to be purchased.

7.5.7 No Relationship Between Tables in the Database

The database tables used for the EDDS do not have relationship among them. To implement a relational database tables consume a lot of time and it is complex. To avoid late system delivery, the database tables used are discrete, because it is much easier to be implemented. Due to this approach, it can be seen that there are a lot of data redundancy in the database. But the data redundancy concept is used because the EDDS treats each requests and feedback received as a unique event.

7.6 Future enhancements

Due to EDDS constraints, some idea have been formulated and suggested by the system adviser, system moderator and also system testing respondents. These suggestions will be the EDDS future enhancements in order to make the EDDS functioning as a real-time system. The EDDS future enhancements are listed below.

7.6.1 Payment Module

One of the most wanted feature for the EDDS in order for it to be functioning as a real time system is the payment module. Payment module is needed to handle all the payment information, which includes payment to the document source and payment from the user who make request. Payment module is a very complex and time consuming to build. Furthermore, a lot of researches need to be carried out to understand the policy of the Clearance Company or financial company about online payment. The payment module will help the system administrator to generate receipt and send payment information the University Malaya Treasury Department. This will save a lot of time because the process is being done electronically and less errors.

7.6.2 More Categorized and Appropriate Presentation of Summaries

The summaries generated by the EDDS need to be more precise in its type of document categorization. The categorization, which was currently implemented by the EDDS, is general. The summaries generated must be able to be categorized for example summaries about request for an article of a journal, request for a chapter of a book and also both at the same time. Various summary presentations also need to be offer to the administrator rather than a bar chart presentation only such as stacked bar, pie chart and line graph. Currently the bar chart presented by the EDDS is in vertical format and is not appropriate because it look too messy. Changes need to be made to so that the bar chart presentation will in on horizontal format.

7.6.3 Source Email Table

Another suggestion of future enhancement is to create a source email table in the database. This table will used to keep all the source email address which has been used before so that the administrator need not to retype the source email every time they process a request. Changes need to be made to the process request module so that it can automatically find relevant email address for the administrator to choose by searching up email address according to the source name. For the email address, which cannot be found in the database, the system must be able to provide the administrator an interface to add the email address of the source into the database.

7.6.4 Attachment/Upload Module

The attachment or upload module need to be engineered so that the create notice module can function as what it is required. Currently the usage of Outlook Express is a temporary solution for the attachment function. There is ASP code, which can make email attachment but it need to be supported by complex VBScript before it can be used or localized for EDDS.

7.6.5 User Portfolio

Another feature suggested is to have user portfolio in the EDDS. User portfolio actually is like a personal directory of the user, which contains user details and preference. The EDDS must be able to recognize the user portfolio and send relevant information or promotion to the user according to their preference. This auto-messaging feature will a value added service for the EDDS.

7.6.6 Task List Layout and Presentation -

The task list layout generated by the EDDS is suggestion only. The task list layout and presentation need to be refined so that it is much useful when the task list is printed out. One of the changes, which need to be made, is to create a remark column in the task list printout so that the administrator can write down their remark about a particular request.

7.6.7 Type of Document Detection

A suggestion has been made to include the document type recognition into the EDDS. Due to time constraint, this features need to be address as future enhancement. The EDDS will have the capability of categorizing the type of user request according to the document title. Currently the user will categorize by them self the type of document for their request. It is hard to know whether the categorization has been made is correct. So it is suggested that the EDDS will categorized the request by examining the document title. The EDDS can do this by matching the keywords in the title with all the related terms in specific discipline and make a suggestion, which type the document it is.

7.6.8 Other Language Support

Currently as what have been stated in the EDDS constraints is the language used by the EDDS as a medium to communicate with the end users. An effort must be carried out to that the EDDS support Bahasa Malaysia.

7.6.9 Online Rules and Regulations for EDDS

There are no rules and regulations been defined and stated for the University of Malaya EDDS. This is due to no such system being implemented by the library. For the EDDS to be implemented as a full working system, the library will have to list out the rules and regulations for this service and publish it online as one of the component in EDDS.

7.7 Knowledge and Experience Gained

By developing the EDDS, personally I myself feel that I have learned a lot of things, which I have never, knew or realized before this. Some of the knowledge and experience gained are listed below.

7.7.1 Web Base Development

Web based system development is different from developing a stand-alone system. The most major difference is the characteristic of the Internet. The Internet is stateless according to what I have learn during my course and I do not have any idea what this concept is until I developed the EDDS. The way to overcome this problem has been implemented from what I have learned from my course. Developing the EDDS is much more like implementing a real working solution based on the theory and knowledge gained.

7.7.2 SQL Command

From the database course, which has been taught, the theory and knowledge of SQL could be implemented in real solution. It is amazing to know that how flexible and powerful SQL command could be by implementing it in real working module. The SQL command taught in classes could now be expanded and applied.

7.7.3 ASP

ASP is not being taught in the courses, but I could now have chance to learn a new language. I have never thought of using ASP is actually so interesting and easy. It offers powerful features in developing a web-based system. This knowledge could be implemented during future projects and works on similar system.

7.7.4 Development Tools

Besides than ASP, I could now use the Visual Interdev to create ASP pages. Interdev seems like not useful to me before I used it to develop the EDDS. It is a great opportunity to know that it can help in speeding up the development process and offers a lot of easy to use features.

7.7.5 JavaScript

JavaScript is different from VBScript. The used of JavaScript is not so important to me before the EDDS development. From the theory and information gathered during development tools analysis, it is said that JavaScript could be implemented in ASP pages. This has been a theory until I have the chance to actually implement JavaScript in the ASP pages during EDDS development.

7.7.6 Developing a Medium Size Web Base System

The EDDS can be considered as a medium size web based system. I have never developed a web-based system before. But eventually, by developing the EDDS, I know how to develop a basic web based system and knows what are the basic requirements needed.

7.7.7 Proper and Well Planned System Development

Software development steps, stages and planning are just a theory to me before I developed the EDDS. But during the development of EDDS, I actually have the chance to practice all the knowledge and theory learned about software development and planning. EDDS was developed using the waterfall methodology, and this is a way to make myself attached to a well-planned schedule rather than developing a system without any planning at all.

7.8 Summary

Chapter 7 listed out the problems encountered and solutions used to solve the problems. Most of the problems could be solved in order to make sure that the EDDS will meet its requirements. Due to time constraint, some problems could not be solved at the moment. Problems that could not be solved have been expressed in the EDDS future enhancements. Future enhancements list out the suggested features; functions and changes need to be implemented if the EDDS would be used as a full working system in the future.

Opinion and advice have been sought from few respondents during the system demo and testing. Their opinions and suggestions have been taken into considerations in making the EDDS more reliable. Some of the suggestions are listed in the future enhancements while some already being implemented.

The EDDS strength and constraints are also being listed out. These strength and constraints are being identified during the EDDS development and evaluation.

By developing the EDDS, I personally myself learned a lot of new programming technique and knowledge. This knowledge can be used in solving future problems and in developing similar system.

7.9 Conclusion

Overall the EDDS project has achieved and fulfills all the requirements and objectives suggested. The significances of the EDDS will be seen if the system could be implemented in a real working environment.

A lot of research and studies have been made during EDDS literature review. All the information gathered has been translated into a solution and could be used for future reference in developing a similar system.

Throughout the EDDS development process, a lot of experience and knowledge gained. And this will help in future effort in developing a web-based system and working aspect. This knowledge gain is very valuable and improved my programming and ability to solve problems.

On the other side of the knowledge gained, from the problems faced and solutions found could be a useful reference in the future because web based systems are gaining their popularities with the advancement of Internet technology.

Appendices

Appendix A

User Manual

Table of Contents

Table of Contents	1
List of Figures	2
List of Tables	2
Part 1: Introduction	
1.1 Introduction	3
1.2 About the Manual	3
Part 2: Hardware and Software Requirements	
2.1 Hardware Requirements	4
2.2 Software Requirements	4
Part 3: User Manual	
3.1 Register As New	5
3.2 Login	8
3.3 Update User Profile	10
3.4 Change Password	11
3.5 Main Menu	12
3.6 Request for an Article of a Journal	13
3.7 Request for a Chapter of a Book	16
3.8 User Feedback	19
3.9 Logout	21

University of Malaya Library

EDDS

User Manual

Table of Contents

Table of Contents	1
List of Figures	2
List of Tables	2
Part 1: Introduction	
1.1 Introduction	3
1.2 About the Manual	3
1.3 Terminologies and Abbreviations Used	3
Part 2: Hardware and Software Requirements	
2.1 Hardware Requirements	4
2.2 Software Requirements	4
Part 3: User Manual	
3.1 Register As New User	5
3.2 Login	8
3.3 Update User Profile	10
3.4 Change Password	11
3.5 Main Menu	12
3.6 Request for an Article of a Journal	13
3.7 Request for a Chapter of a Book	16
3.8 User Feedback	19
3.9 Logout	21

List of Figures

Figures	Description	Page Num.
Figure 1	Login Screen	5
Figure 2	Registration Screen	6
Figure 3	Successful Registration Notification and Request Menu	7
Figure 4	Login Screen	8
Figure 5	User Profile Screen	9
Figure 6	Update User Profile Screen	10
Figure 7	Change Password Screen	11
Figure 8	Main Menu	12
Figure 9	Request for an Article of a Journal Form	13
Figure 10	Request for an Article of a Journal Verification Screen	14
Figure 11	Request for an Article of a Journal Confirmation Screen	15
Figure 12	Request for a Chapter of a Book Form	16
Figure 13	Request for a Chapter of a Book Verification Screen	17
Figure 14	Request for a Chapter of a Book Confirmation Screen	18
Figure 15	User Feedback Form	19
Figure 16	User Feedback Confirmation	20
Figure 17	Logout Screen	21

List of Table

Table	Description	Page Num.
Table 1	Terminologies and Abbreviations Used	3

Part 1: Introduction

1.1 Introduction

Welcome to the University of Malaya Electronic Document Delivery Services system (EDDS). The EDDS promises a new perspective and evolution in the University of Malaya Library Document Delivery Services.

The Electronic Document Delivery System (EDDS) is a web-based system that promises a quick and efficient way to access information for University of Malaya. The entire request can be made electronically and the materials shall be sent to the requester electronically. The service is also available 24 hours per day throughout the year because it is available online. The EDDS will cater the request for an article of a journal or a chapter of a book, which is not found in the University of Malaya Library collections.

This manual will provide a step-by-step comprehensive guide for the user of this system. Each step will be elaborated in detail together with the appropriate screen shot of the system.

1.2 About the Manual

This user manual is organized into 3 different parts as listed below:

- Part 1: The Introduction
- Part 2: Hardware and Software Requirements
- Part 3: The User Manual

1.3 Terminologies and Abbreviations Used

Num.	Term/Abbreviations	Meaning
1.	User ID	User Identification
2.	Request	Either request for an article of a journal or request for a chapter of a book
3.	You	End user
4.	Redirect	Link to another web page
5.	New User	User which never have been register to use the system before
6.	Link	A hyperlink on the web page
7.	Icon	A graphical representation of a link which can be click

Table 1: Terminologies and Abbreviations Used

Part 2: Hardware and Software Requirements

2.1 Hardware Requirements

The hardware required for this system:

- i. Minimum Pentium II 233Mhz
- ii. Minimum 64MB Ram Memory
- iii. 4GB hard disk
- iv. Modem
- v. Phone line

2.1 Software Requirements

The software required for this system:

- i. Internet Explorer or Netscape Navigator
- ii. Outlook express or any email program



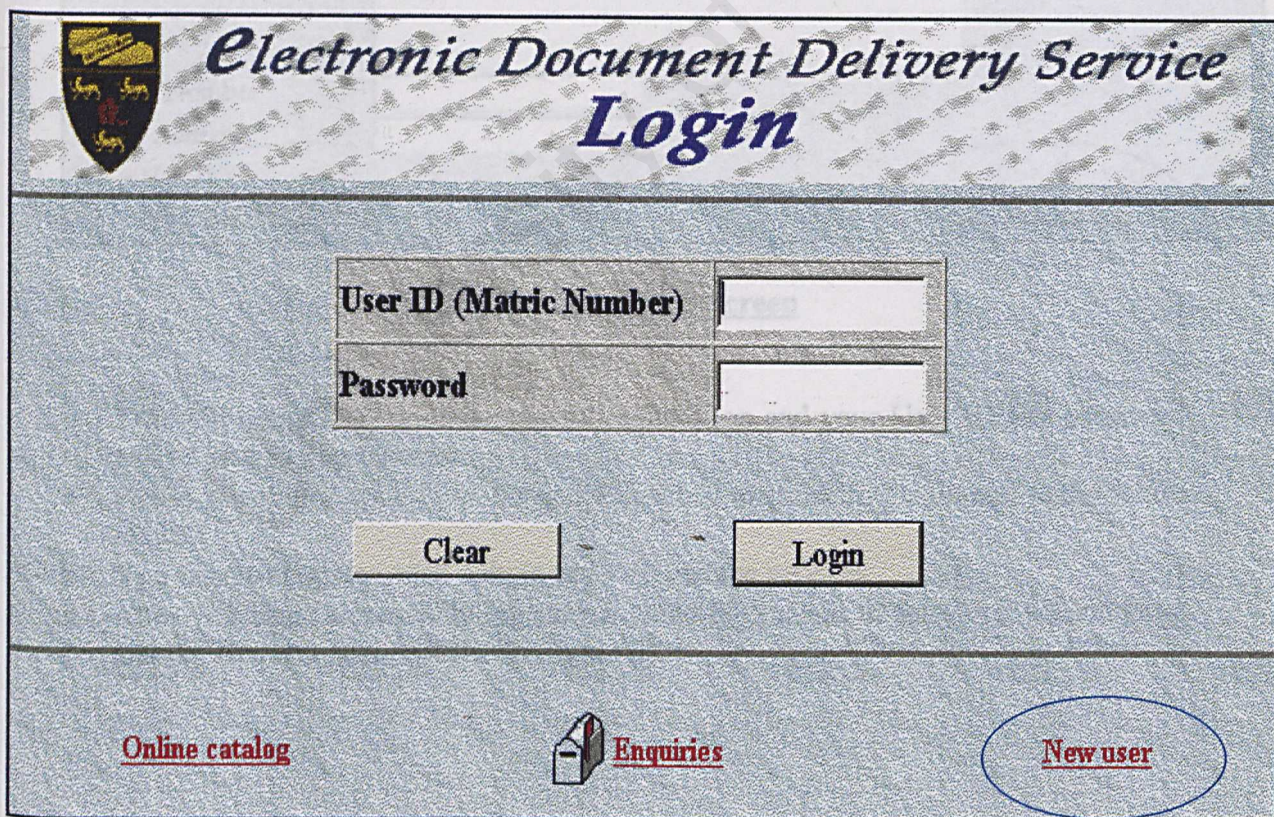
Part 3: User Manual

This manual content the description of the user functions, which are:

1. Register as New User
2. Login
3. Update User profile
4. Change Password
5. Main Menu
6. Request for an Article of a Journal
7. Request for a Chapter of a Book
8. User Feedback
9. Logout

3.1. Register As New User

- If you have never register before, you will be required to register with the system before you can proceed with any request.
- On the Login Screen, select the **New User** link at the bottom right corner. (Refer to Figure 1)



Electronic Document Delivery Service
Login

User ID (Matric Number)	<input type="text"/>
Password	<input type="password"/>


[Online catalog](#)  [Enquiries](#) [New user](#)

Figure 1 Login Screen

- You will be directed to the Registration screen.
- Enter your personal particulars and other related fields.
- Make note that some fields, are compulsory to fill in, and some are optional.
- Then click the Submit button to continue or Clear button to clear all fields, which you have just entered. (Refer to Figure 2)

**Electronic Document Delivery System
Registration**

Name *

User ID(Matric Number) *

User Password * (Password must be between 4 to 10 characters long)

Reconfirm Password *

Semester Address *

Mailing Address *

Contact Number *

E-mail Address *

* Required field

Figure 2 Registration Screen

- You will be notified upon successful registration and your User ID and password will be sent to you by email
- Please remember your User ID and Password in order to Login and make future request.
- You need to register once only in order to use the services
- After you have successfully registered, you will directed to the Confirmation Page and the Request Menu (Refer to Figure 3)

3.2. Login

Enter your User ID and Password at the Login Screen before you can use the services

Click the Login button (middle right of the Login Screen) to login or the Clear button (middle left of the Login Screen) to clear all the fields on the Login Screen


Congratulations! You have succesfully registered.

Your personal particulars:

User ID	BEE000450
Name	Chiew Pek Hui
Semester	Kolej Kediaman Tun Ahmad Zaidi Universiti Malaya
Address	59300 Lembah Pantai Kuala Lumpur
Mailing Address	2352, Jalan RS 3/8 Taman Rasah Jaya 70300 Negeri Sembilan
Contact Number	012-66695423
eMail Address	cp_hui@usa.net

[Request for an Article of a Journal](#)

[Request for a chapter of a Book](#)

[Update Profile](#)

[Change Password](#)

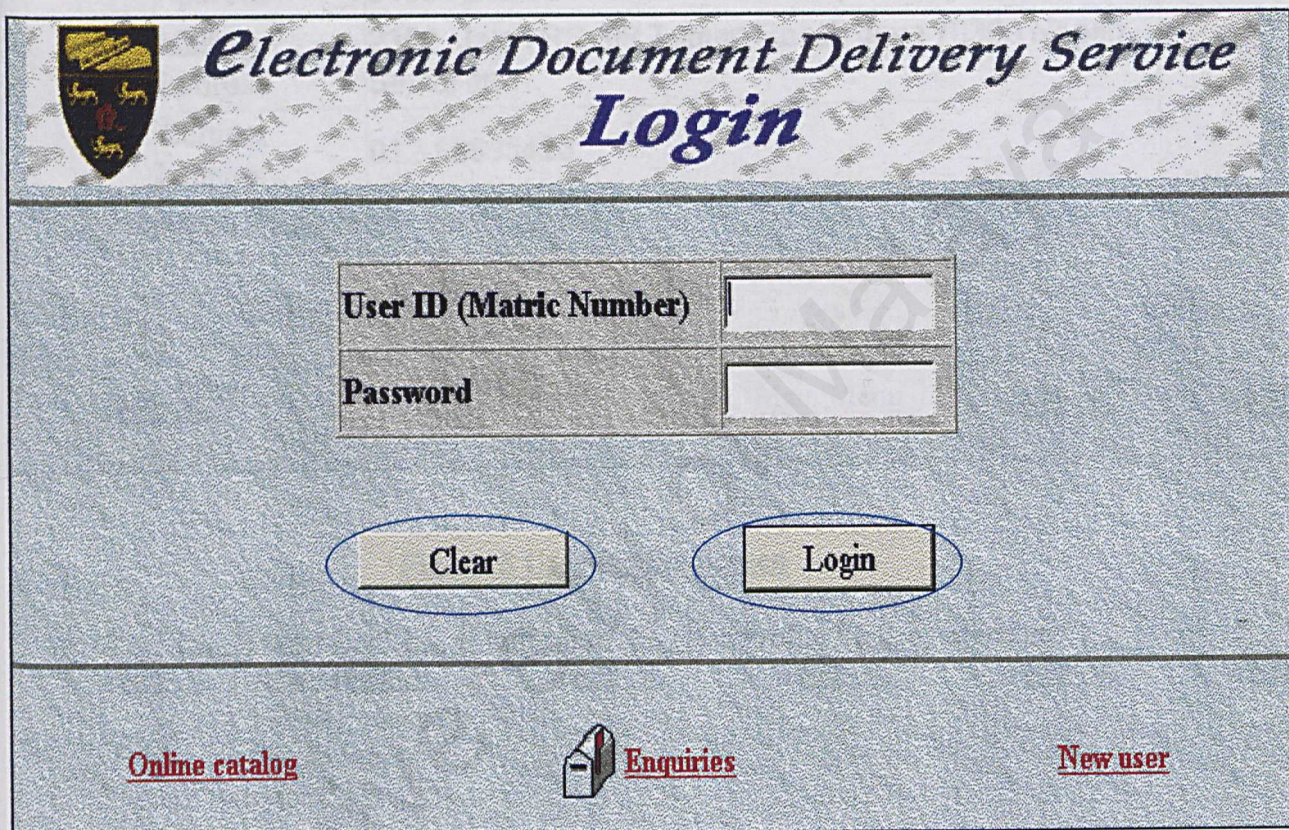
[User Feedback](#)

[Logout](#)

Figure3 Successful Registration Notification and Request Menu

3.2. Login

- Enter your User ID and Password at the Login Screen before you can use the services
- Click the Login button (middle right of the Login Screen) to login or the Clear button (middle left of the Login Screen) to clear all the fields on the Login Screen (Refer to Figure 4)
- You will be prompted to re-enter your User ID and Password if you typed in wrongly



Electronic Document Delivery Service
Login


User ID (Matric Number)	<input type="text"/>
Password	<input type="password"/>

[Clear](#) [Login](#)

[Online catalog](#) [Enquiries](#) [New user](#)


Figure 4 Login Screen


- After you have Login successfully you be directed to the User Profile Screen (Refer to Figure 5)
- There are 2links available at the Login Screen which are
 - i. Online Catalog (University of Malaya Library OPAC)
 - ii. Enquiries (if you have any question to ask)
 - iii. New User (to register, for new user only)




Electronic Document Delivery System User Profile

User ID	BEE000450
Name	Chiew Pek Hui
Semester Address	Kolej Kediaman Tun Ahmad Zaidi Universiti Malaya 59300 Lembah Pantai Kuala Lumpur
Mailing Address	2352, Jalan RS 3/8 Taman Rasah Jaya 70300 Negeri Sembilan
Contact Number	012-66695423
eMail Address	cp_hui@usa.net


[Update Profile](#)


[Change Password](#)


[Main Menu](#)



[Logout](#)


Figure 5 User Profile Screen

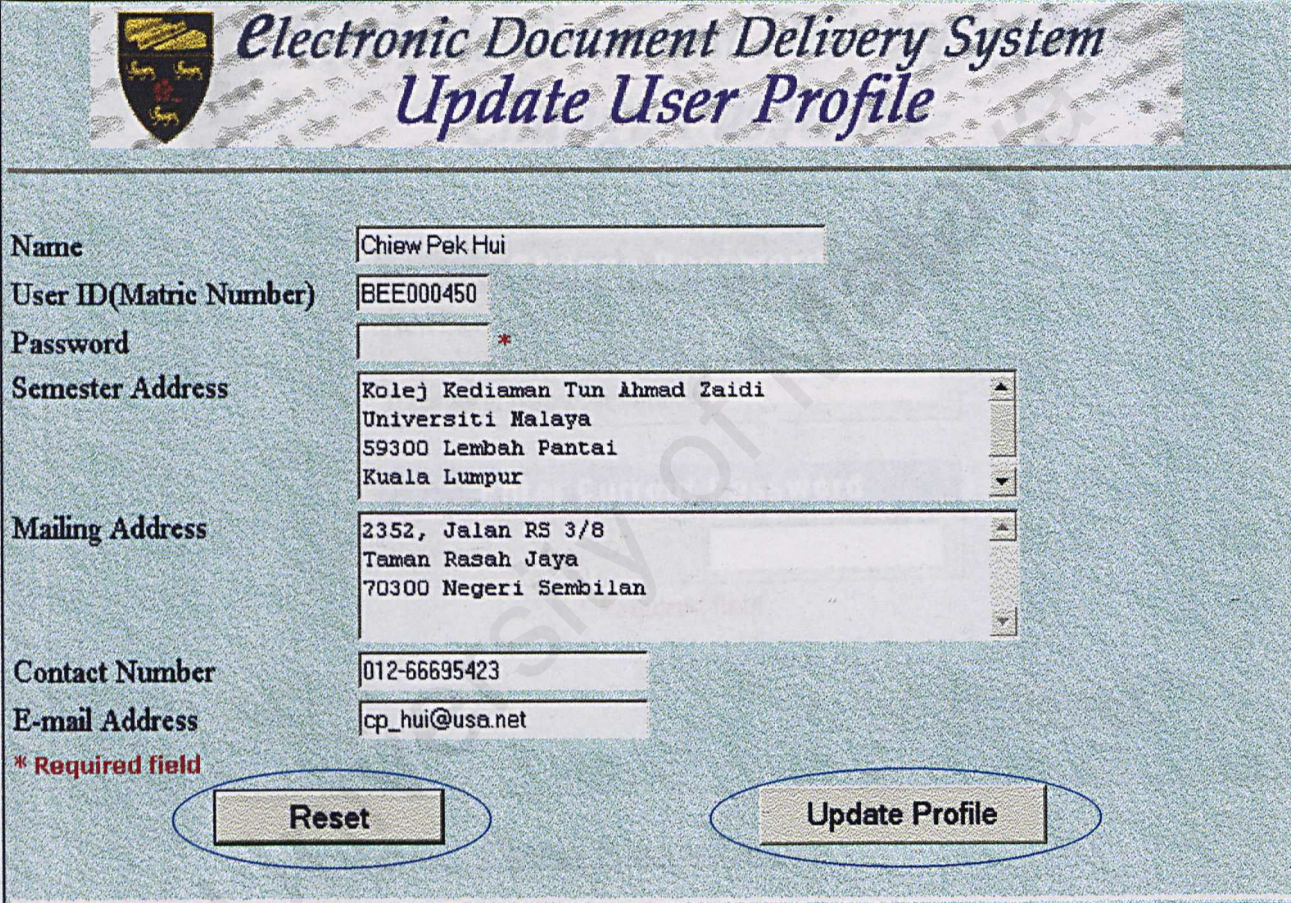
- There are 4 links available at bottom of the User Profile Screen, which are
 - i. Update Profile
 - ii. Change Password
 - iii. Main Menu
 - iv. Logout

Figure 5 User Profile Screen

3.3 Update User Profile



- If you wish to update your personal profile, click the Update Profile icon 
- You can change or append your personal details on the Update User Profile Screen (Refer to Figure 6)
- Enter your password before you click Update Profile button (on the bottom right of screen) to update your profile (Refer to Figure 6)
- Click the Reset Button (on the bottom left of screen) if you want cancel all the changes you have made (Refer to Figure 6)



Electronic Document Delivery System
Update User Profile

Name: Chiew Pek Hui

User ID(Matric Number): BEE000450

Password: *

Semester Address: Kolej Kediaman Tun Ahmad Zaidi
Universiti Malaya
59300 Lembah Pantai
Kuala Lumpur

Mailing Address: 2352, Jalan RS 3/8
Taman Rasah Jaya
70300 Negeri Sembilan

Contact Number: 012-66695423

E-mail Address: cp_hui@usa.net

* Required field

Reset Update Profile

Figure 6 Update User Profile Screen

3.4 Change Password



- If you wish to change your user password, click the Change Password icon.
- Key-in your new password and re-confirm your new password
- Enter your current password before you click Change Password button (on the bottom right of screen) to change to your new password (Refer to Figure 7)
- Click the Reset Button (on the bottom left of screen) if you want to clear all the fields on the Change (Refer to Figure 7)



Electronic Document Delivery System
Change Password

Change Password

New Password*
(4 to 10 characters long)

Reconfirm New Password*

Enter Current Password

Password*

* Required field

Reset

Change Password

Figure 7 Change Password Screen

3.5 Main Menu



- To continue with your request click on the Main Menu icon.
- You will be directed to the Main Menu (Refer to Figure 8)
- There are 6 links available at the Main Menu, which are
 - i. Request for an Article of a Journal
 - ii. Request for a chapter of a Book
 - iii. Update Profile
 - iv. Change Password
 - v. User Feedback
 - vi. Logout
- The Update Profile, Change Password and Logout icons and links have the same function as what have been describe earlier

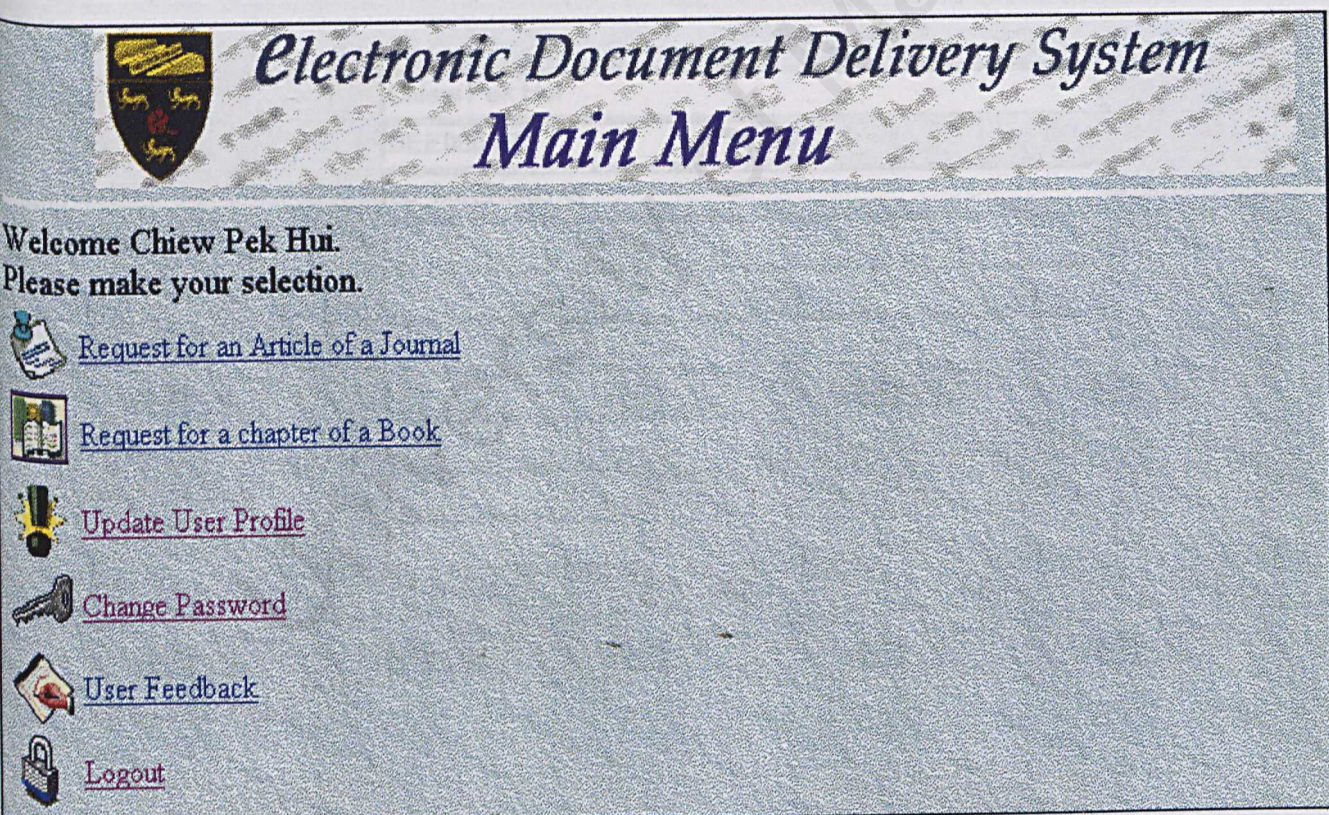



Figure 8 Main Menu

3.6 Request for an Article of a Journal



- To make a request for an article of a journal, click the Request for an Article of a Journal icon 
- You will be redirected to the request for an article of a journal form (Refer to Figure 9)
- Fill up the form fields according to your request
- Click the Submit Request button (on the bottom right) to submit your request (Refer to Figure 9)
- Click the Reset button (on the bottom left) to clear all the fields in the form (Refer to Figure 9)



Electronic Document Delivery System
Article of a Journal

Name	Chiew Pek Hui		
User ID	BEE000450		
eMail	cp_hui@usa.net		
Journal Title			
Article Author			
Article Title			
Publish Date	* mm/dd/yy Format		
Volume			
Issue Number			
Pages of Articles			
Type	<input checked="" type="radio"/> Science <input type="radio"/> Medical <input type="radio"/> Architecture <input type="radio"/> Computer <input type="radio"/> Economy <input type="radio"/> Sociology <input type="radio"/> Engineering <input type="radio"/> Law <input type="radio"/> Other		

* Required field

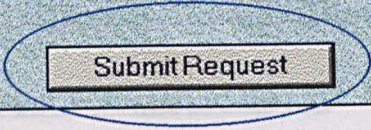




Figure 9 Request for an Article of a Journal Form

- You will be redirected to the request for an article of a journal verification screen (Refer to Figure 10)
- Check all the details on screen and make sure all the details are correct
- If the details are not correct, hit the back button on your web browser and rectify the errors and click the Submit Request button again. You will be directed again to the request for an article of a journal verification screen.
- Click the Confirm Request button if there is no changes need to be made




- Click on the Cancel Request icon (on bottom left)  to cancel the current request



Electronic Document Delivery System

Article of a Journal

Name	Chiew Pek Hui
User ID	BEE000450
eMail	cp_hui@usa.net
Journal Title	Journal of Documentation
Article Author	Robertson, S.E.
Article Title	The parametric description of retrieval tests: part 1: the basic parameters
Publish Date	01/01/1969
Volume	25
Issue Number	-
Pages of Articles	1-27
Type	Computer

 **Cancel Request**

Confirm Request

Figure 10 Request for an Article of a Journal Verification Screen

- After confirming your request, you will be redirected to the request for an article of a journal confirmation screen (Refer to Figure 11)
- There are 4 links available at bottom of the request for an article of a journal confirmation screen, which are
 - i. Main Menu
 - ii. Another Request (to make another request for an article of a journal)
 - iii. Logout
 - iv. Feedback

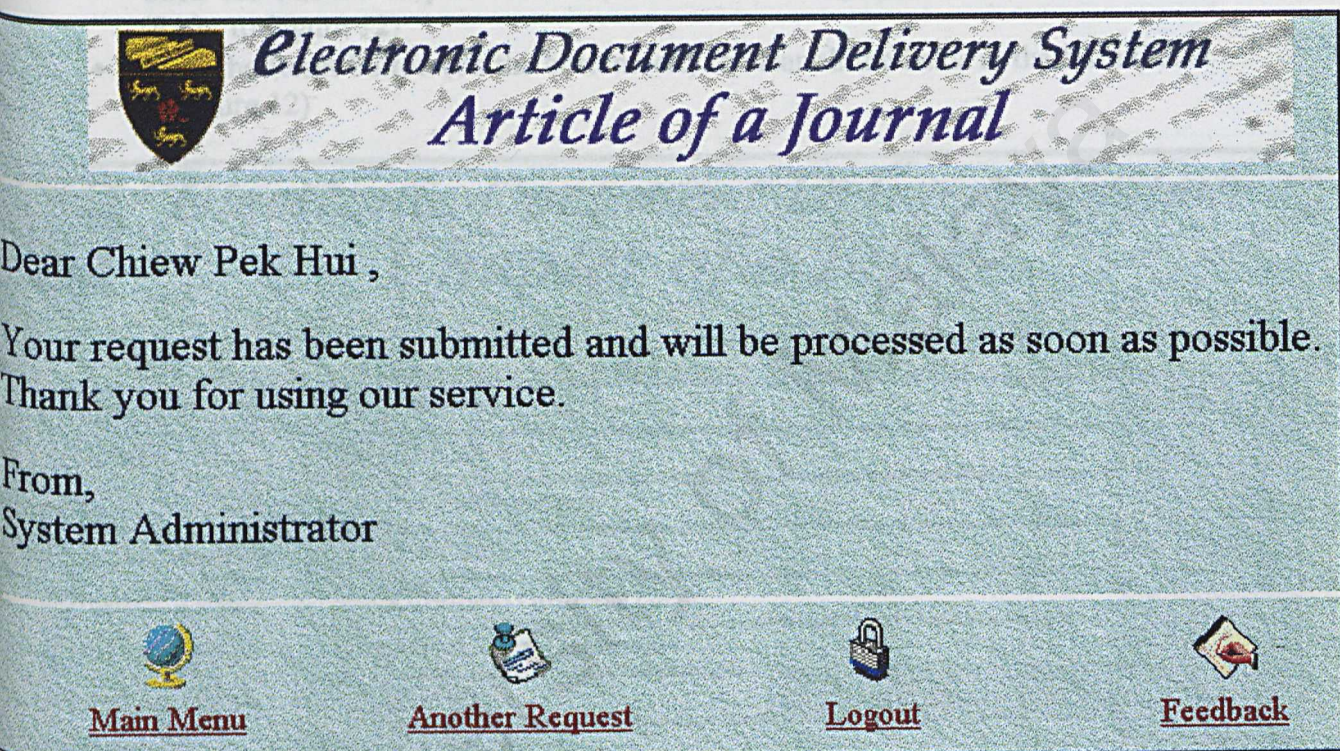




Figure 11 Request for an Article of a Journal Confirmation Screen

3.7 Request for a Chapter of a Book



- To make a request for a Chapter of a Book, click the Request for a Chapter of a Book icon 
- You will be redirected to the request for a Chapter of a Book form (Refer to Figure 12)
- Fill up the form fields according to your request
- Click the Submit Request button (on the bottom right) to submit your request (Refer to Figure 12)
- Click the Reset button (on the bottom left) to clear all the fields in the form (Refer to Figure 12)



Electronic Document Delivery System

Chapter of a Book

Name	Chiew Pek Hui		
User ID	BEE000450		
eMail	cp_hui@usa.net		
Chapter Author	Ed Tittel *		
Chapter Title	Grabbing Graphics Gusto *		
Title of Book	HTML 4 For Dummies *		
Pages in Book	85-100		
Publisher	IDG Books *		
Date	01/01/2000 * mm/dd/yy Format		
Place Publish	Foster City		
ISBN	0-7645-0723-0		
Type	<div><div>Please select one of the appropriate type</div><div><div><input type="radio"/> Science</div><div><input checked="" type="radio"/> Computer</div><div><input type="radio"/> Engineering</div></div><div><div><input type="radio"/> Medical</div><div><input type="radio"/> Economy</div><div><input type="radio"/> Law</div></div><div><div><input type="radio"/> Architecture</div><div><input type="radio"/> Sociology</div><div><input type="radio"/> Other</div></div></div>		

* Required field


Reset


Submit Request

Figure 12 Request for a Chapter of a Book Form

- You will be redirected to the request for a Chapter of a Book verification screen (Refer to Figure 13)
- Check all the details on screen and make sure all the details are correct
- If the details are not correct, hit the back button on your web browser and rectify the errors and click the Submit Request button again. You will be directed again to the request for a Chapter of a Book verification screen.
- Click the Confirm Request button if there is no changes need to be made




- Click on the Cancel Request icon (on bottom left)  to cancel the current request



Electronic Document Delivery System Chapter of a Book

Name	Chiew Pek Hui
User ID	BEE000450
eMail	cp_hui@usa.net
Chapter Author	Ed Tittel
Chapter Title	Grabbing Graphics Gusto
Title of Book	HTML 4 For Dummies
Pages in Book	85-100
Publisher	IDG Books
Date	01/01/2000
Place Publish	Foster City
ISBN	0-7645-0723-0
Type	Computer

 **Cancel Request**

Confirm Request

Figure 13 Request for a Chapter of a Book Verification Screen

- After confirming your request, you will be redirected to the request for a Chapter of a Book confirmation screen (Refer to Figure 14)
- There are 4 links available at bottom of the request for a Chapter of a Book confirmation screen, which are
 - i. Main Menu
 - ii. Another Request (to make another request for a Chapter of a Book)
 - iii. Logout
 - iv. Feedback



Electronic Document Delivery System *Chapter of a Book*

Dear Chiew Pek Hui,

Your order has been submitted and will be processed as soon as possible.
Thank you for using our service.

From,
System Administrator



[Main Menu](#)



[Another Request](#)



[Logout](#)




[Feedback](#)

Figure 14 Request for a Chapter of a Book Confirmation Screen

3.8 User Feedback



- To post your feedback or question, click the Feedback icon
- You will be redirected to the Feedback form (Refer to Figure 15)
- Fill up the feedback form
- Click the Submit button (on the bottom right) to submit your feedback (Refer to Figure 15)
- Click the Reset button (on the bottom left) to clear your feedback (Refer to Figure 15)



Electronic Document Delivery System
User Feedback

User ID

me

Email Address

User Feedback

BEE000450

Chiew Pek Hui

cp_hui@usa.net

Reset

Submit

Figure 15 User Feedback Form

- After submitting your feedback, you will be redirected to the Feedback confirmation screen (Refer to Figure 16)
- There are 2 links available at bottom of the Feedback Confirmation Screen, which are
 - i. Main Menu
 - ii. Logout



Dear Chiew Pek Hui
Your feedback will be process by our staff.

Thank you for submitting your feedback and the effort to improve this service!



[Main Menu](#)


[Logout](#)

Figure 16 User Feedback Confirmation

9. Logout



- Click the Logout icon to logout from the system after you have finish using the system
- 
- Remember to logout before you leave the system to avoid unauthorized person to use your account.
- You will be redirected to the Logout Screen when you have successfully logout from the system (Refer to Figure 17)

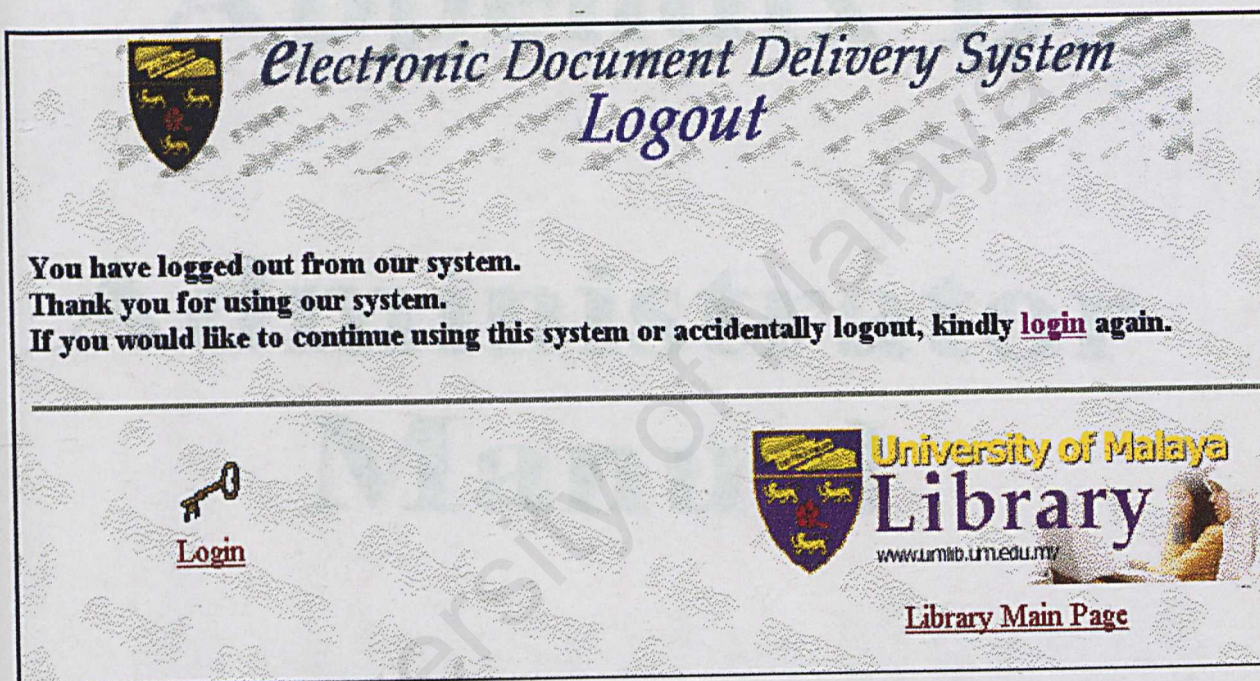


Figure 17 Logout Screen

- To Login again click on the Login icon



- To go to the University of Malaya Library Main Page click the link on the bottom right of the screen

Appendix B

Administrator

Manual

Table of Contents

Table of Contents	1
List of Figures	2
List of Tables	2
Part 1: Introduction	
1.1 Introduction	3
1.2 About the Manual	3
1.3 Terminologies and Abbreviations Used	3
Part 2: Hardware and Software Requirements	
2.1 Hardware Requirements	4
2.2 Software Requirements	4
Part 3: Administrator	
3.1 Login	5
3.2 Administrator Main Menu	6
3.3 Generate Task list	7
3.4 Print	7
3.5 Help	7
3.6 Create Notice	17
3.7 Create Notice	17
3.8 Create Pick-up Notice	21
3.9 Logout	25

University of Malaya Library
EDDS
Administrator Manual

Table of Contents

Table of Contents	1	5
List of Figures	2	6
List of Tables	2	7
Part 1: Introduction		8
1.1 Introduction	3	9
1.2 About the Manual	3	10
1.3 Terminologies and Abbreviations Used	3	11
Part 2: Hardware and Software Requirements		12
2.1 Hardware Requirements	4	13
2.2 Software Requirements	4	14
Part 3: Administrator Manual		15
3.1 Login	5	16
3.2 Administrator Main Menu	6	17
3.3 Generate Task list	7	18
3.4 Process Request	9	19
3.5 Reply User Feedback	13	20
3.6 Generate Summaries	15	22
3.7 Create Notice	17	23
3.8 Create Pickup Notice	21	24
3.9 Logout	25	

List of Figures

Figures	Description	Page Num.
Figure 1	Login Screen	5
Figure 2	Administrator Main Menu	6
Figure 3	Generate Task List Screen	7
Figure 4	Generated Task List	8
Figure 5	Process Request Screen	9
Figure 6	Process a Request page	10
Figure 7	Source and Source email fields	11
Figure 8	Process Confirmation Page	12
Figure 9	Reply User Feedback Form	13
Figure 10	Save Reply for the Feedback Confirmation Page	14
Figure 11	Generate Summaries Form	15
Figure 12	Generated Summaries	16
Figure 13	Create Notice Screen	17
Figure 14	User and Request Details	18
Figure 15	Email Program Invoked	19
Figure 16	Create Pickup Notice Screen	20
Figure 17	User and Request Details	22
Figure 18	Send Pickup Notice Confirmation Page	23
Figure 19	Logout Screen	24

List of Table

Table	Description	Page Num.
Table 1	Terminologies and Abbreviations Used	3

Table 1. Terminologies and Abbreviations Used

Part 1: Introduction

1.1 Introduction

Welcome to the University of Malaya Electronic Document Delivery Services system (EDDS). The EDDS promises a new perspective and evolution in the University of Malaya Library Document Delivery Services.

The Electronic Document Delivery System (EDDS) is a web-based system that promises a quick and efficient way to access information for University of Malaya. The entire request can be made electronically and the materials shall be sent to the requester electronically. The service is also available 24 hours per day throughout the year because it is available online. The EDDS will cater the request for an article of a journal or a chapter of a book, which is not found in the University of Malaya Library collections.

This manual will provide a step-by-step comprehensive guide for the system administrator of this system. Each step will be elaborated in detail together with the appropriate screen shot of the system.

1.2 About the Manual

This user manual is organized into 3 different parts as listed below:

- Part 1: The Introduction
- Part 2: Hardware and Software Requirements
- Part 3: The Administrator Manual

1.3 Terminologies and Abbreviations Used

Num.	Term/Abbreviations	Meaning
1.	Administrator ID	Administrator Identification
2.	Request	Either request for an article of a journal or request for a chapter of a book
3.	You	Administrator
4.	Redirect	Link to another web page
5.	User	End-user who is using the service
6.	Link	A hyperlink on the web page
7.	Icon	A graphical representation of a link which can be click

Table 1: Terminologies and Abbreviations Used

Part 2: Hardware and Software Requirements

2.1 Hardware Requirements

The hardware required for this system:

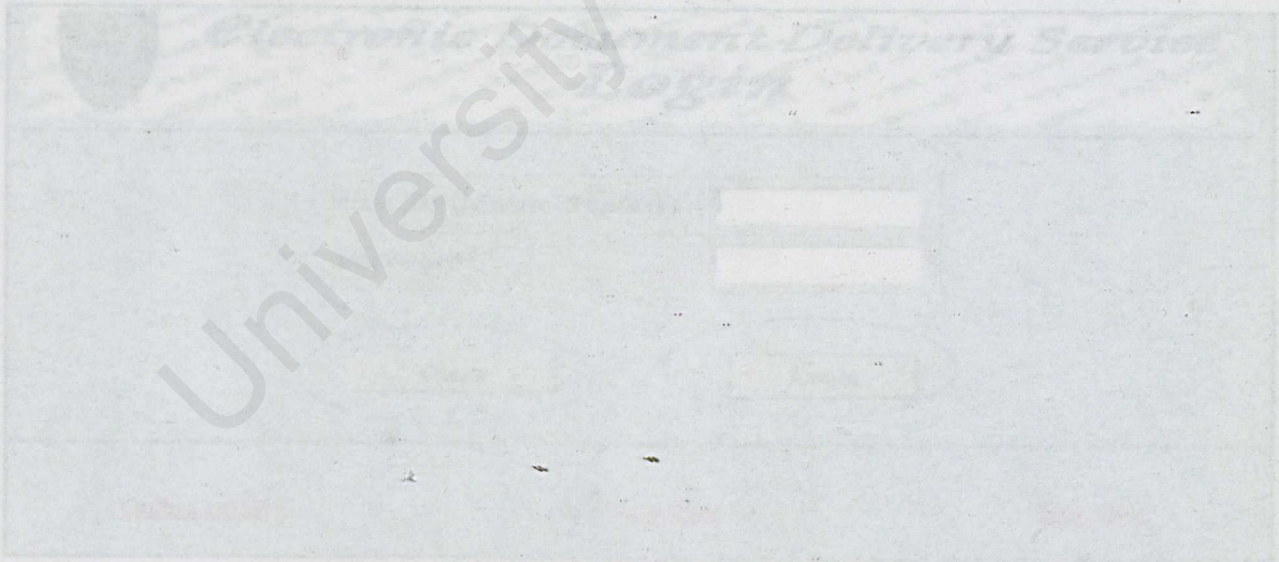
- i. Minimum Pentium III or compatible PC based computers with 500MHz Processor
- ii. Minimum 256MB Ram Memory
- iii. 10GB hard disk
- iv. Modem
- v. Phone line

2.1 Software Requirements

The software required for this system:

- i. Internet Explorer or Netscape Navigator
- ii. Outlook express or any email program
- iii. Windows 2000 Server or Windows 2000 Advance Server

- Click the Login button (on bottom right) to login to the system (Refer to Figure 1)
- Click the Clear button (on bottom left) to clear all the fields in the Login screen (Refer to Figure 1)
- If you have enter the wrong Administrator ID or Password, the system will prompt you to re-enter the Administrator ID and Password



- You will be directed to the Administrator Page. Please login as administrator

Part 3: Administrator Manual

This manual content the description of the administrator functions as listed below:

1. Login
2. Administrator Main Menu
3. Generate Task List
4. Process Request
5. Reply User Feedback
6. Generate Summary
7. Create Notice
8. Create Pickup Notice
9. Logout

3.1 Login

- At the Login screen, type in your Administrator ID in the User ID field
- Enter your Administrator Password
- Click the Login button (on bottom right) to login to the system (Refer to Figure 1)
- Click the Clear button (on bottom left) to clear all the fields in the Login screen (Refer to Figure 1)
- If you have enter the wrong Administrator ID or Password, the system will prompt you to re-enter the Administrator ID and Password

**Electronic Document Delivery Service
Login**

User ID (Matric Number)

Password


[Online catalog](#)  [Enquiries](#) [New user](#)

Figure 1 Login Screen

- You will be directed to the Administrator Main Menu once you have successfully login as administrator

3.2 Administrator Main Menu

- There are 7 links available at the Main Menu (Refer to Figure 2) which are
 - i. Generate Task List
 - ii. Process Request
 - iii. Reply User Feedback
 - iv. Generate Summary
 - v. Create Notice
 - vi. Create Pickup Notice
 - vii. Logout

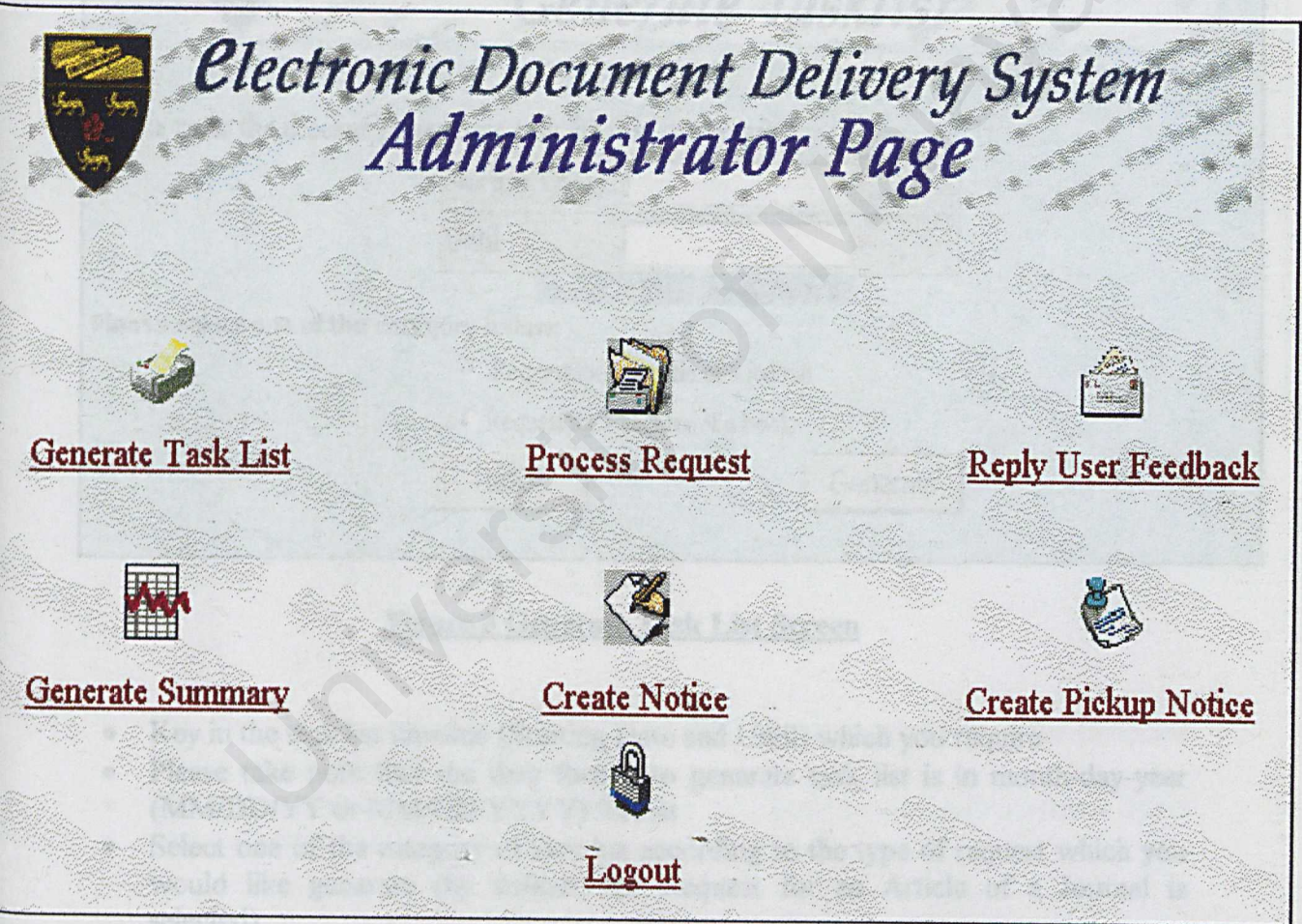


Figure 2 Administrator Main menu

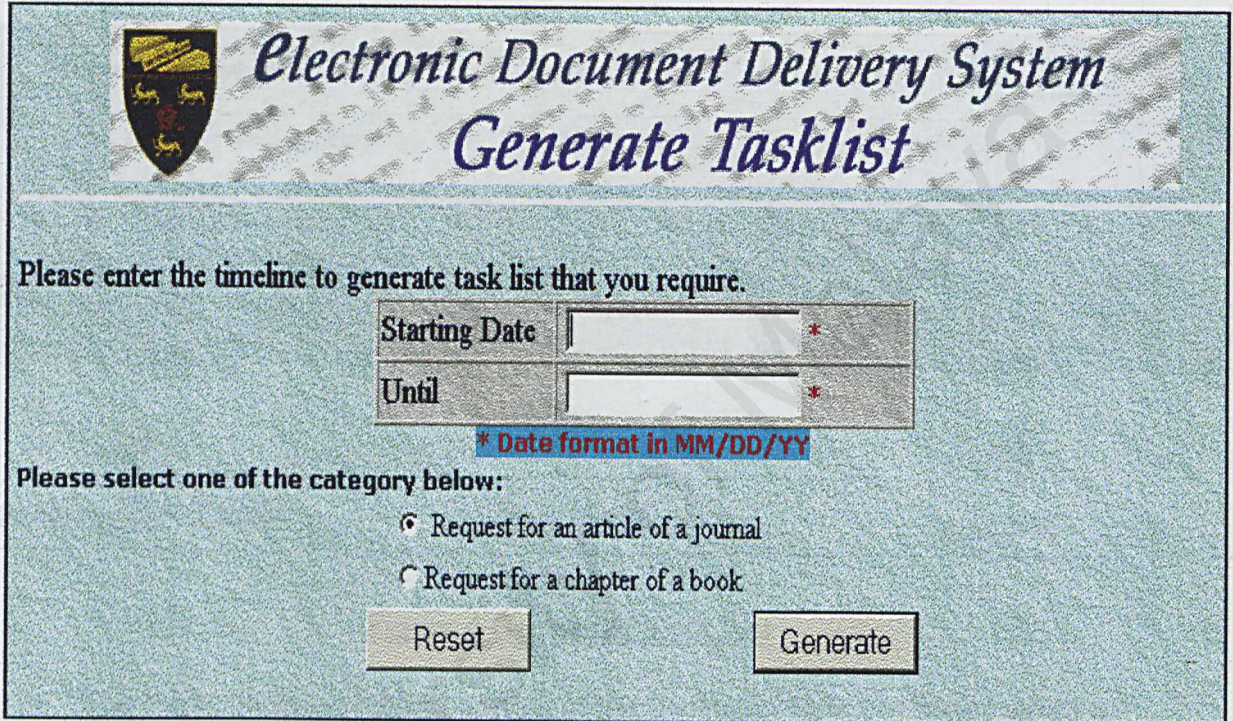
3.3 Generate Task List



- To generate task list of the requests received which have not been processed, click on the Generate Task List icon



- You will be redirected to the Generate Task List screen (Refer to Figure 3)



Electronic Document Delivery System
Generate Tasklist

Please enter the timeline to generate task list that you require.

Starting Date *

Until *

* Date format in MM/DD/YY

Please select one of the category below:

☒ Request for an article of a journal

☐ Request for a chapter of a book

Reset Generate

Figure 3 Generate Task List Screen

- Key in the task list timeline (Starting Date and Until) which you require
- Please take note that the date format to generate task list is in month-day-year (MM/DD/YY or MM/DD/YYYY) format
- Select one of the category of task list according to the type of request which you would like generate (by default, the Request for an Article of a Journal is selected)
- Click the Generate button to generate the required task list
- Click the Reset button clear all the fields in the Generate task List screen
- The required task list will be display after all the input are verified correct (Refer to Figure 4)
- Should there be any error in the task list display, please check the timeline so that it is valid

- Only the request that have not been processed will be listed in the task list

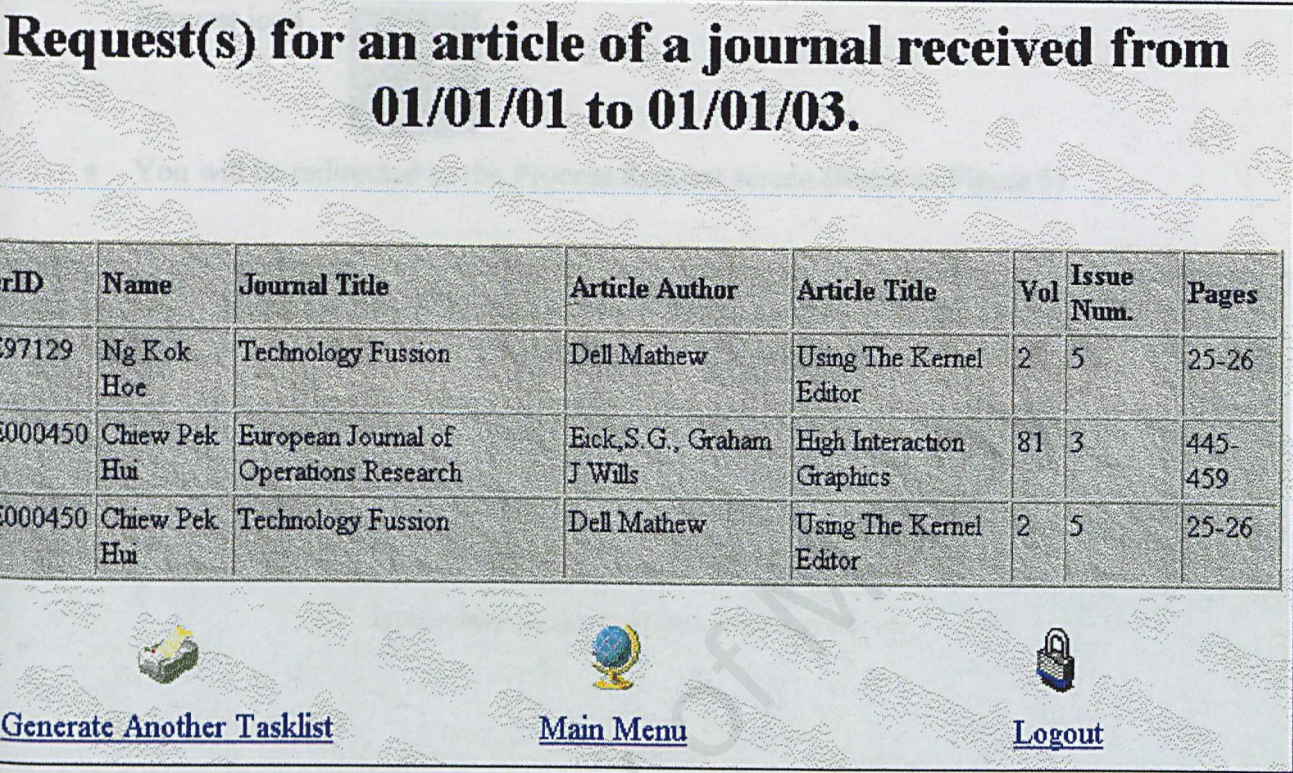


Figure 4 Generated Task List

- There are 3 links available at bottom of the Generated Task List screen which are
 - Generate Another Task List
 - Main Menu
 - Logout
- The Generate Another Task List link and icon will bring you back to the Generate Task List screen (Refer to Figure 3)
- The Main Menu link and icon will bring you back to the Administrator Main Menu (Refer to Figure 2)

3.4 Process Request



- To process any request which have not been processed, click on the Process Request icon



- You will be redirected to the Process Request screen (Refer to Figure 5)

Electronic Document Delivery System
Process Request

Please select one of the category below:

- ☐ Process request(s) for an article of a journal
- ☐ Process request(s) request for a chapter of a book


Process

[Main Menu](#)

[Logout](#)

Figure 5 Process Request Screen

- Choose one of the two category which you wish to process
- Click the Process button to start process the particular type of request
- You will redirected to the processing page (Refer to Figure 6)
- The processing page will list out all the details about a particular user request
- The requests which have not been processed are listed according to first come first serve basis
- You will have to do the searching for the material using any search engine or contact you have
- Type in the Source name and the Source Email address which have the require document (Refer to Figure 7)




Electronic Document Delivery System


Article of a Journal

Name	Ng Kok Hoe
User ID	EEE97129
eMail	nhk1976@hotmail.com
Journal Title	Technology Fussion
Article Author	Dell Mathew
Article Title	Using The Kernel Editor
Publish Date	5/6/2001
Volume	2
Issue Number	5
Pages of Articles	25-26
Source	
Source email	

Reset

Process

[Main Menu](#)

[Process Request Menu](#)


[Logout](#)

Figure 6 Process a Request page

- There are 3 links located at the bottom of the Process a Request page. The three links are
 - i. Main Menu (if you wish to stop processing any request including the current request shown, and return to main menu)
 - ii. Process Request Menu (if you wish to process other type of request (request for a chapter of a book or request for an article of a journal)
 - iii. Logout

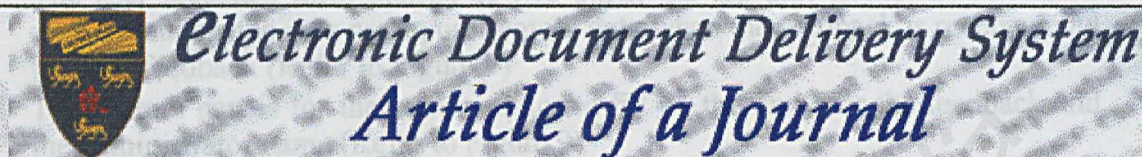
Name	Ng Kok Hoe
User ID	EEE97129
eMail	nrk1976@hotmail.com
Journal Title	Technology Fussion
Article Author	Dell Mathew
Article Title	Using The Kernel Editor
Publish Date	5/6/2001
Volume	2
Issue Number	5
Pages of Articles	25-26
Source	
Source email	

Figure 7 Source and Source email fields

*Type in the Source and the
Source email address here*

- Click the Process button (on bottom left) to process the request when you have type in the required Source name and the Source Email address (Refer to Figure 6)
- Once the request is being processed, the system will send an email to the source using the Source mail that you have given. A notice on successful processing will be email to the respective requesting user.
- You will be directed to the Process Confirmation page after the email and notice being sent (Refer to Figure8)

- Process confirmation page will list out the source name and the source email address which are being used to send the request email and also the reference number for the particular request (Refer to Figure 8)
- Reference numbers are label according to the type of request. Reference number starting with the letter J, indicate that the particular request is a request for an article of a journal while reference number starting with the letter C, indicate that the particular request is a request for a chapter of a book
- Click the Reset button (on bottom left) to clear the Source and Source email fields in the process request page (Refer to Figure 6)



The request has been sent to **University of Malaya** at **library@um.edu.my** with our reference **J2**.



[Main Menu](#)



[Process Another Request](#)



[Logout](#)

Figure 8 Process Confirmation Page


- There are 3 links available at bottom of the Process Confirmation Page, which are
 - i. Main Menu (return to main menu)
 - ii. Process Another Request (return to the Process Request screen)
 - iii. Logout

3.5 Reply User Feedback



- If you wish to read through and reply user feedback about the services and system, click the Reply User Feedback icon
- You will be directed to the reply user feedback form (Refer to Figure 9)
- The user feedback form consist of two parts, the top part is the user feedback which contains the user name, user email, date when the feedback is received and the user feedback (Refer to Figure 9)
- The bottom part is for the administrator to fill up the reply for the feedback and the administrator name (Refer to Figure 9)





Electronic Document Delivery System
Reply User Feedback


Name	Chiew Pek Hui
Email	cp_hui@usa.net
Date	1/20/2002
Feedback	Please notify me when my request is ready for collection.

Reply Feedback

Reply By

Reset

Submit


[Main Menu](#)



[Logout](#)

Figure 9 Reply User Feedback Form

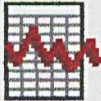
- Click the Reset button (on bottom left) to clear the Reply Feedback and Reply By fields in the reply user feedback form (Refer to Figure 9)
- Click the Submit button (on bottom left) to submit and save your reply to the user feedback (Refer to Figure 9)
- When your reply to the user feedback has been successfully saved, you will be redirected to the save reply for the feedback confirmation page. At this point, a copy of the reply will be sent to the user through user email address (Refer to Figure 10)



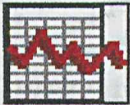
Figure 10 Save Reply for the Feedback Confirmation Page

- There are 3 links available at bottom of the Process Confirmation Page, which are
 - i. Reply Another Feedback (return to the Reply User Feedback form)
 - ii. Main Menu (return to main menu)
 - iii. Logout
- Should all the user feedback being replied; the system will prompt you with a message saying that all feedback has been replied.


3.6 Generate Summaries



- If you wish to generate summaries about the service, click the Generate Summaries icon.



- You will be directed to the Generate Summaries form (Refer to Figure 11)



Electronic Document Delivery System Summaries

Please enter the timeline to generate summary that you require.

Starting Date	<input type="text"/>	*
Until	<input type="text"/>	*

* Date format in MM/DD/YY

Figure 11 Generate Summaries Form

- Key in the summaries timeline (Starting Date and Until) which you require
- Please take note that the date format to generate summaries is in month-day-year (MM/DD/YY or MM/DD/YYYY) format
- Click the Generate button to generate the required summaries
- Click the Reset button clear all the fields in the Generate Summaries form
- The required summaries will be display after all the input are verified correct (Refer to Figure 12)
- Should there be any error in the summaries display, please check the timeline so that it is valid
- The summaries will take into consideration all the requests received in between the timeline, regardless the request has been processed or not

3.7 Create Notice

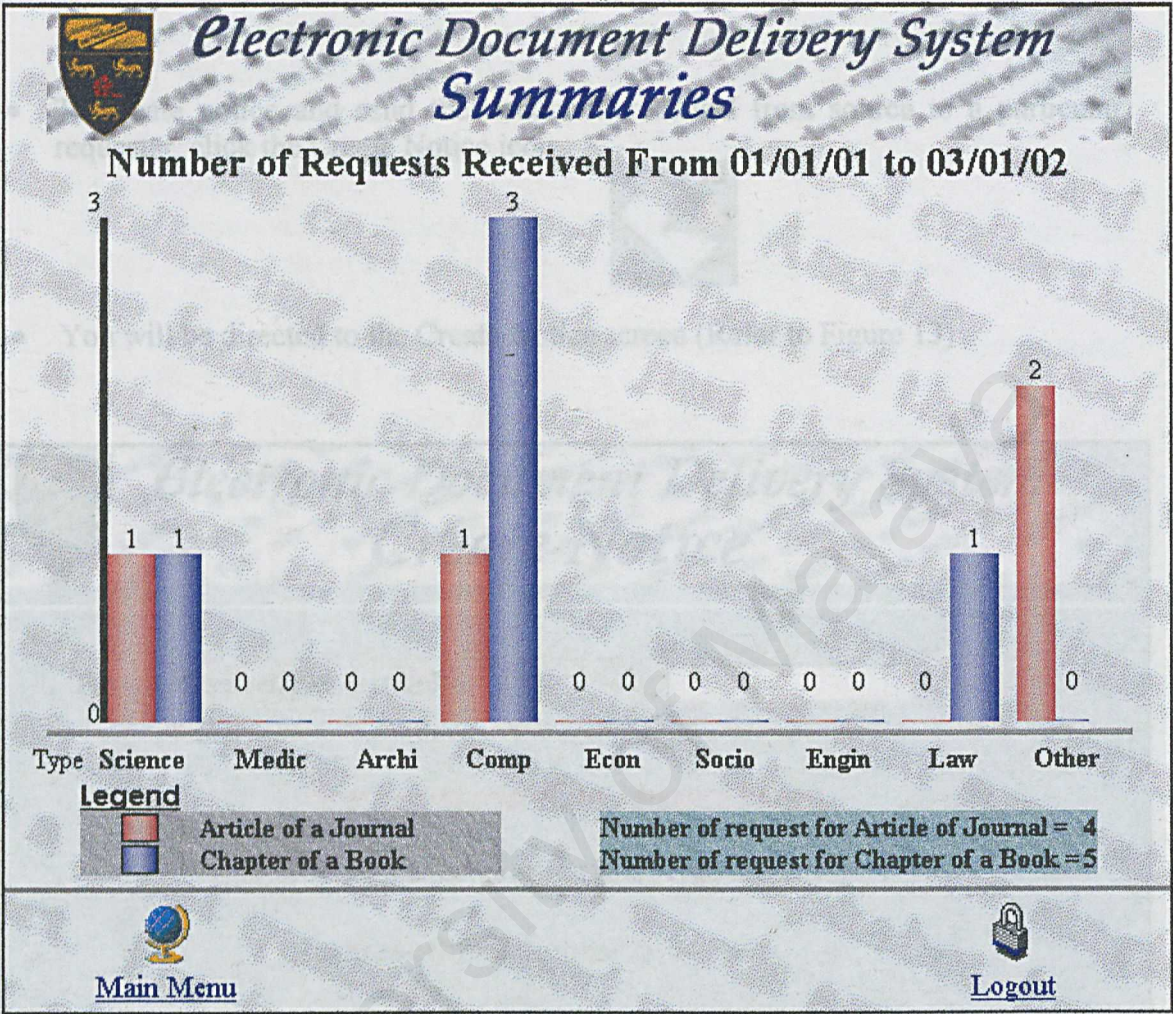


Figure 12 Generated Summaries

3.7 Create Notice



- To create notice and send the document received from source to a particular requester, click the Create Notice icon



- You will be directed to the Create Notice screen (Refer to Figure 13)

Electronic Document Delivery System
Create Notice

Please enter reference number

Please select one of the category below:

☐ Request for an article of a journal

☐ Request for a chapter of a book



 [Main Menu](#) [Logout](#) 

Figure 13 Create Notice Screen

- Enter the reference number of the document to track back the particular requester

- The reference number is according to the reference number use when sourcing the document during processing.
- The reference number should be the one stated in the email reply by the document in their reply email. Should the reference number is not there, please contact the database administrator to perform search to locate the particular requester
- Select one of the request categories, request for an article of a journal or request for a chapter of a book
- The reference number indicates category. Reference number starting with the letter J is an article of a journal while reference number starting with the letter C is a chapter of a book
- Click the Submit button (on bottom right) to locate the search for the requester record (Refer to Figure 13)
- The requester and the request details will be shown if the record is found on the user and request detail page (Refer to Figure 14). Please check all the request details is correct before sending the document to the requester
- If the particular record is not found, the system will notify you
- Click the Reset button (on bottom right) to clear all the fields in the Create Notice Screen (Refer to Figure 13)
- There are 3 links available at bottom of the User and Request Detail page (Refer to Figure 14), which are
 - i. Main Menu (return to main menu)
 - ii. Create Other Notice (return to the Create Notice screen)
 - iii. Logout

Figure 14: User and Request Details

- Click on the email link or the letter “HERE” from the sentence “Click HERE to notify user” which is located at the bottom of the detail table (Refer to Figure 14) to activate the email program (Refer to Figure 15)
- Type in your email content and click on the Attach button (Refer to Figure 14) to attach the particular document to be sent to the requester
- Click the send button (Refer to Figure 15)

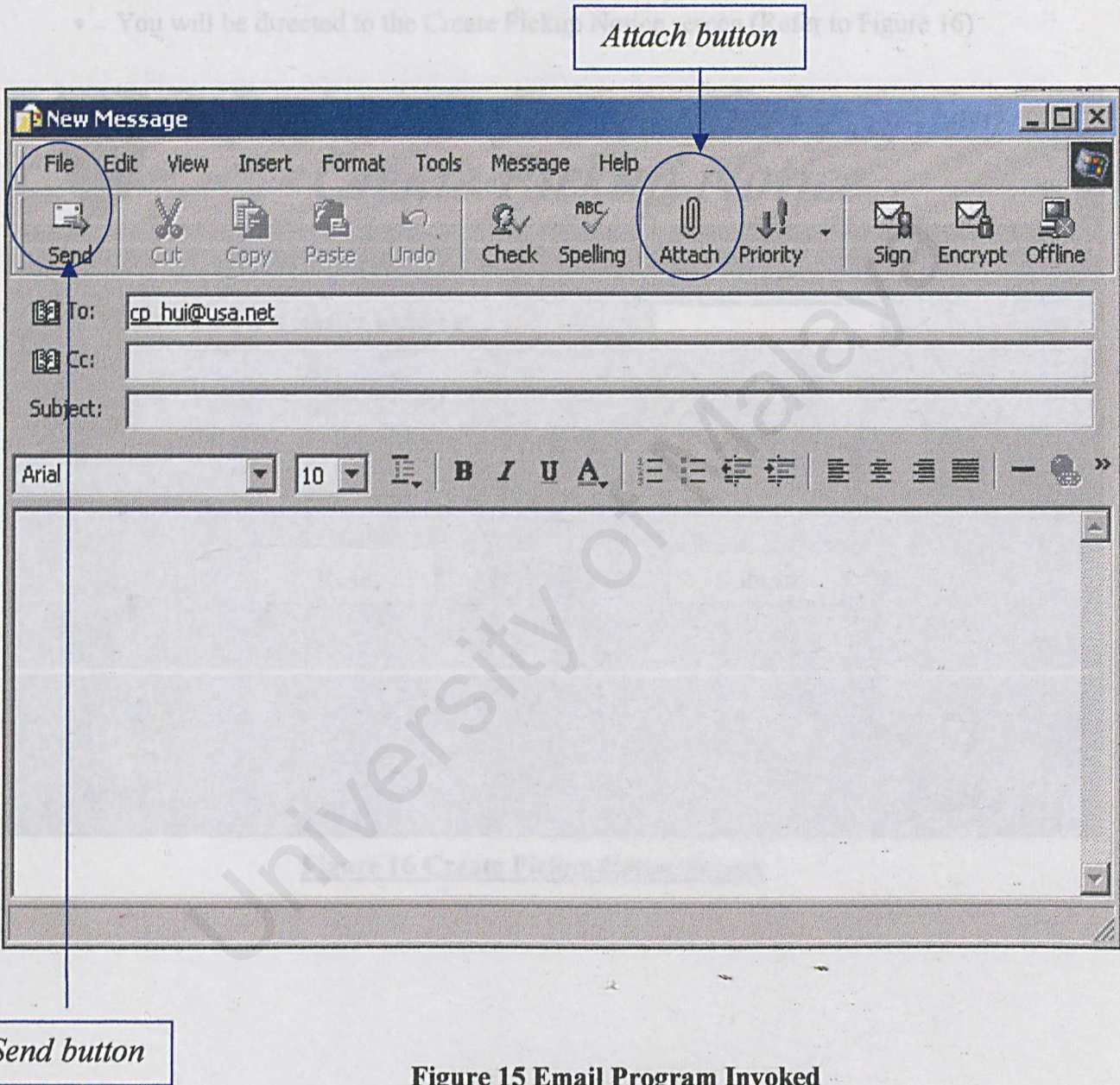



Figure 15 Email Program Invoked

- After the notice and the document being sent to the user, close the email program and click on one of the 3 available links below the User and Request Details page

3.8 Create Pickup Notice



- If the materials or documenta arrived from the source are not in electronic form, click the Create Pickup Notice icon to notify the requester to come and collect the materials or document from the library counter
- You will be directed to the Create Pickup Notice screen (Refer to Figure 16)



Electronic Document Delivery System

Create Pickup Notice


Please enter reference number

Please select one of the category below:

- ☐ Request for an article of a journal
- ☐ Request for a chapter of a book

Reset

Submit

[Main Menu](#)


[Logout](#)

Figure 16 Create Pickup Notice Screen

- Enter the reference number of the document to track back the particular requester
- The reference number is according to the reference number use when sourcing the document during processing
- The reference number should be the one stated in the email reply by the document in their reply email. Should the reference number is not there, please contact the database administrator to perform search to locate the particular requester
- Select one of the request categories, request for an article of a journal or request for a chapter of a book
- The reference number indicates category. Reference number starting with the letter J is an article of a journal while reference number starting with the letter C is a chapter of a book
- Click the Submit button (on bottom right) to locate the search for the requester record (Refer to Figure 16)
- The requester and the request details will be shown if the record is found on the user and request detail page (Refer to Figure 17). Please check all the request details is correct before sending the pickup notice to the requester
- If the particular record is not found, the system will notify you
- Click the Reset button (on bottom right) to clear all the fields in the Create Notice Screen (Refer to Figure 16)
- There are 3 links available at bottom of the User and Request Detail page (Refer to Figure 17), which are
 - i. Main Menu (return to main menu)
 - ii. Create Other Pickup Notice (return to the Create Pickup Notice screen)
 - iii. Logout



Electronic Document Delivery System

Create Pickup Notice

Name	Chiew Pek Hui
User ID	BEE000450
eMail	cp_hui@usa.net
Journal Title	European Journal of Operations Research
Article Author	Eick S.G., Graham J Wills
Article Title	High Interaction Graphics
Publish Date	3/1/1999
Volume	81
Issue Number	3
Pages of Articles	445-459
Source	National University of Malaysia
Source email	library@ukm.edu.my

Collection Date	<input type="text"/>	*
Collection Time	<input type="text"/>	*
Extra notices	<input type="text"/>	
Reply By	<input type="text"/>	*

Reset

Send Pickup


[Main Menu](#)


[Create other pickup notice](#)


[Logout](#)

Figure 17 User and Request Details

- Type in the collection date and collection time you would like the requester to come to the library counter to collect the document (Refer to Figure 17)
- The pickup notice is a standard notice which contains the collection date, collection time and the name of the person who wrote the pickup notice
- If you would like to add other extra information besides than those mention above, fill up the Extra Notices field (Refer to Figure 17)
- Click the Send Pickup button (on bottom right) to send out the pickup notice to the requester (Refer to Figure 17)
- You will be redirected to the Send Pickup Notice Confirmation page upon successful pickup notice delivery (Refer to Figure 18)

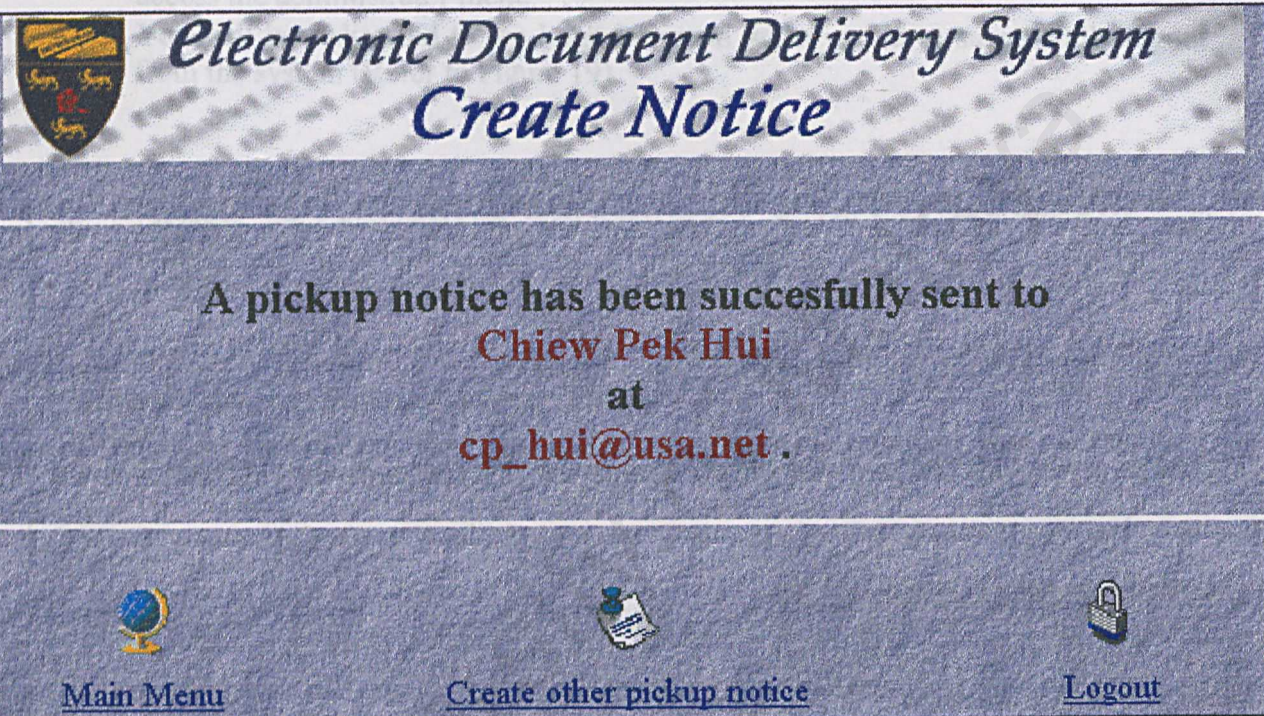


Figure 18 Send Pickup Notice Confirmation Page

- There are 3 links available at bottom of the Send Pickup Notice Confirmation Page (Refer to Figure 18), which are
 - i. Main Menu (return to main menu)
 - ii. Create Other Pickup Notice (return to the Create Pickup Notice screen)
 - iii. Logout

3.8 Logout



- Click the Logout icon to logout from the system after you have finish using the system



- Remember to logout before you leave the system to avoid unauthorized person to access the administrator menu.
- You will be redirected to the Logout Screen when you have successfully logout from the system (Refer to Figure 19)

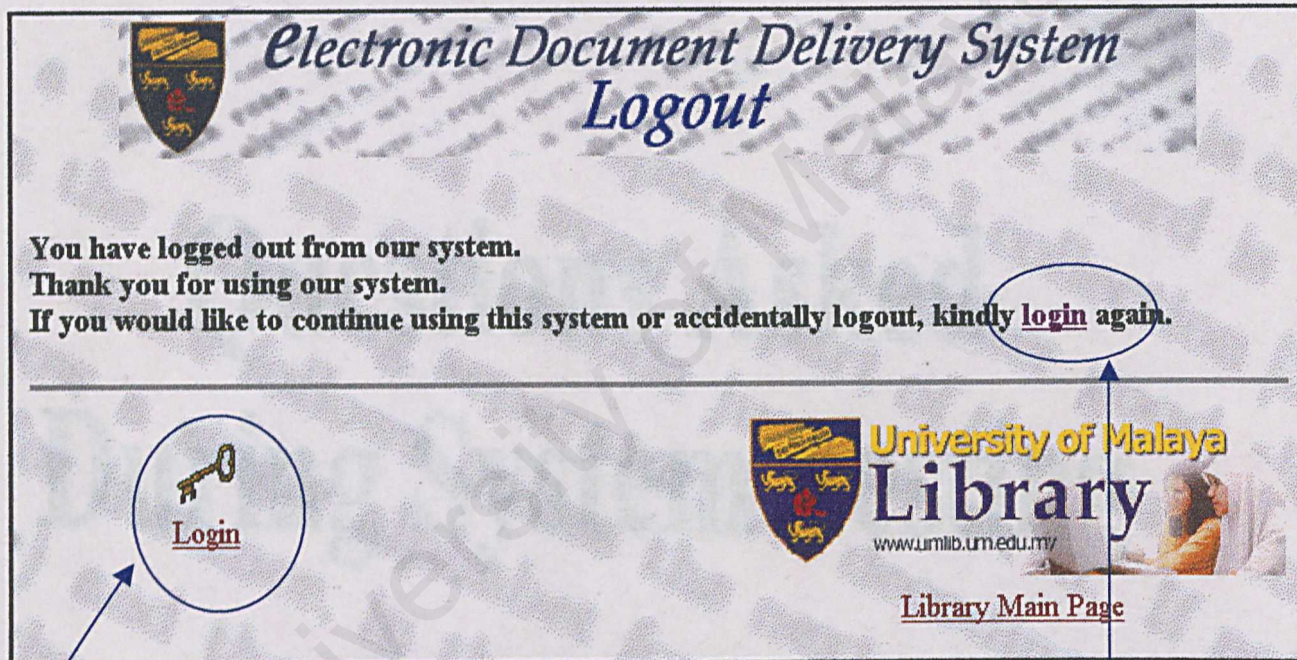


Figure 19 Logout Screen

Login icon

Login link

- To Login again click on word “login” in the “If you would like to continue using this system or accidentally logout, kindly login again” sentence or the Login icon (Refer to Figure 19)



- To go to the University of Malaya Library Main Page click the link on the bottom right of the screen

Below are several questions asked during system testing. The respondents are the system adviser, the system moderator, and students from Computer Science and Information Technology Faculty and also from other faculties.

1. What do you think about the overall interface design?
2. Are all the functions provided useful?
3. Do you have any suggestions in adding or modifying the EDDS?
4. What is your overall opinion about the EDDS?
5. Is the system easy to use?
6. Do you find any difficulty to use the EDDS?
7. Do you need a long time to understand how to use the EDDS?

Appendix C

Questions Asked During System Testing

Below are several questions asked during system testing. The respondents are the system adviser, the system moderator, and students from Computer Science and Information Technology Faculty and also from other faculties.

1. What do you think about the overall interface design?
2. Are all the functions provided useful?
3. Do you have any suggestions in adding extra functions for the EDDS?
4. What is your opinion about the response time?
5. Is the system user-friendly?
6. Do you find any difficulty to use the EDDS?
7. Do you need a long time to understand how to use the EDDS?

Below are the sample coding for EDDS. Only few modules coding are shown.

1. Request for an Article of a Journal

```
<%@ Language=VBScript %>
<%
    Session("Journal Title")= ""
    Session("Journal ArticleAuthor")= ""
    Session("Journal ArticleTitle")= ""
    Session("Journal PublishDate")= ""
    Session("Journal Vol")= ""
```

Appendix D

```
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio .NET">
<script language="javascript">
```

```
function validateForm() {
```

Coding

```
var JournalTitle=document.getElementById("JournalTitle").value;
```

```
var ArticleAuthor=document.getElementById("ArticleAuthor").value;
```

```
var ArticleTitle=document.getElementById("ArticleTitle").value;
```

```
var PublishDate=document.getElementById("PublishDate").value;
```

```
if (JournalTitle=="") { alert("Journal Title field is empty. Please fill in.");
```

```
document.getElementById("JournalTitle").focus();
```

```
return false;
```

```
}
```

```
if (ArticleAuthor=="") { alert("Please enter the Article's Author");
```

```
document.getElementById("ArticleAuthor").focus();
```

```
return false;
```

```
}
```

```
if (ArticleTitle=="") { alert("Article Title field is empty. Please fill in.");
```

```
document.getElementById("ArticleTitle").focus();
```

```
return false;
```

```
}
```

```
if (PublishDate=="") { alert("Please type in the Publish Date");
```

```
document.getElementById("PublishDate").focus();
```

```
return false;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
</script>
```

```
</HEAD>
```

```
<BODY topmargin=0 onLoad = "document.getElementById("JournalTitle").focus();
```

```
background="Pic/Background_Req.gif">
```


Below are the sample coding for EDDS. Only few modules coding are shown.

1. Request for an Article of a Journal

```
<%@ Language=VBScript %>
<%
    Session("Journal_Title")= ""
    Session("Journal_ArticleAuthor")= ""
    Session("Journal_ArticleTitle")= ""
    Session("Journal_PublishDate")= ""
    Session("Journal_Vol")= ""
    Session("Journal_Issue")= ""
    Session("Journal_ArticlePage")= ""
    Session("Journal_Type")= ""
%>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
<script language="javascript">

function validateForm() {
    var JournalTitle=document.JournalForm.JournalTitle.value
    var ArticleAuthor=document.JournalForm.ArticleAuthor.value
    var ArticleTitle=document.JournalForm.ArticleTitle.value
    var PublishDate=document.JournalForm.PublishDate.value

    if(JournalTitle==""){ alert("Journal Title field is empty. Please fill in")
        document.JournalForm.JournalTitle.focus()
        return false
    }
    if(ArticleAuthor==""){ alert("Please type in the Article's Author")
        document.JournalForm.ArticleAuthor.focus()
        return false
    }
    if(ArticleTitle==""){ alert("Article Title field is empty. Please fill in")
        document.JournalForm.ArticleTitle.focus()
        return false
    }

    if(PublishDate==""){ alert("Please type in the Publish Date")
        document.JournalForm.PublishDate.focus()
        return false
    }
}

</script>
</HEAD>
<BODY topmargin=0 onLoad = "document.JournalForm.JournalTitle.focus()"
background="Pic\Background_Reg.gif">
```



```
<CENTER><IMG src="pic/journal.gif"></CENTER><HR size=3 color=white>

<FORM method="post" name="JournalForm" action="VerJournal.asp">

<TABLE background="Pic\Background.gif" align=center border=1 cellPadding=0
cellSpacing=0 id=TABLE1 width="608" style="WIDTH: 608px; HEIGHT: 263px">
<TR>
<TD><FONT size=4>Name</FONT></TD>
<TD><INPUT id=text1 name=text1 readOnly style="HEIGHT: 22px; WIDTH: 350px"
value="<%=Session("CurrentUserName")%>"
</TD>
</TR>
<TR>
<TD><FONT size=4>User ID</FONT></TD>
<TD><INPUT id=text2 name=text2 readOnly style="HEIGHT: 22px; WIDTH: 180px"
value="<%=Session("CurrentUserID")%>" </TD></TR>
<TR>
<TD><FONT size=4>eMail</FONT></TD>
<TD><INPUT id=text1 name=text1 readOnly
style="HEIGHT: 22px; WIDTH: 350px" value="<%=Session("CurrentUserEmail")%>"
</TD></TR>
<TR>
<TD>
<FONT size=4>Journal Title</FONT>
</TD>
<TD>
<FONT size=4>
<INPUT name=JournalTitle style="HEIGHT: 22px; WIDTH: 350px" tabIndex=1
></FONT><FONT color=red>*</FONT></TD> </TR>
<TR>
<TD>
<FONT size=4>Article Author</FONT>
</TD>
<TD>
<FONT size=4><INPUT name=ArticleAuthor style="HEIGHT: 22px; WIDTH: 350px"
tabIndex=2></FONT><FONT
color=red>*</FONT>
</TD>
</TR>
<TR>
<TD>
<FONT size=4>Article Title</FONT>
</TD>
<TD><FONT size=4><INPUT
name=ArticleTitle style="HEIGHT: 22px; WIDTH: 350px" tabIndex=3
></FONT><FONT
color=red>*</FONT>
</TD> </TR>
<TR>
<TD>
```



```
<FONT size=4>Publish Date</FONT>
</TD>
<TD><FONT size=4>
<INPUT name=PublishDate style="HEIGHT: 22px; WIDTH: 180px" tabIndex=4>
<FONT color=red size=3> * mm/dd/yy Format</FONT></FONT>
</TD> </TR>

<TR> <TD>
<FONT size=4>Volume</FONT>
</TD>
<TD>
<FONT size=4>
<INPUT name=Vol style="HEIGHT: 22px; WIDTH: 180px" tabIndex=5></FONT>
</TD> </TR>

<TR> <TD>
<FONT size=4>Issue Number</FONT>
</TD>
<TD>
<FONT size=4><INPUT
name=Issue style="HEIGHT: 22px; WIDTH: 120px" tabIndex=6></FONT>
</TD> </TR>

<TR> <TD>
<FONT size=4>Pages of Articles</FONT></TD>
<TD>
<FONT size=4>
<INPUT name=ArticlePage style="HEIGHT: 22px; WIDTH: 120px" tabIndex=7 ></FONT>
</TD> </TR>

<TR> <TD valign=top>
<FONT size=4>Type
<BR><FONT color=red face=Verdana size=2>Please select one<BR> of the appropriate
type</FONT></FONT> </FONT>
</TD>
<TD> <FONT size=3>
<TABLE cellSpacing=0 cellPadding=0 width="413" border=0 id=inner_table >

<TR> <TD><INPUT type=radio CHECKED value=Science name=type>Science</TD>
<TD><INPUT type=radio value=Medical name=type>Medical</TD>
<TD><INPUT type=radio value=Architecture name=type>Architecture </TD></TR>

<TR>
<TD><INPUT type=radio value=Computer name=type>Computer</TD>
<TD><INPUT type=radio value=Economy name=type>Economy</TD>
<TD><INPUT type=radio value=Sociology name=type>Sociology</TD></TR>
<TR>
<TD><INPUT type=radio value=Engineering name=type>Engineering</TD>
<TD><INPUT type=radio value=Law name=type>Law</TD>
<TD><INPUT type=radio value=Other name=type>Other</TD></TR>
```


</TABLE></TD></TR></TABLE>

<TABLE align=center border=0 cellPadding=1 cellSpacing=1 id=TABLE1 width="80%" style="HEIGHT: 14px; WIDTH: 595px">

<TR>
<TD>* Required
field </TD></TR></TABLE>

<TABLE align=center border=0 cellPadding=1 cellSpacing=1 width="100%" style="HEIGHT: 11px; WIDTH: 526px">

<TR>
<TD width="33%"><INPUT name=reset1 type=reset value=Reset tabIndex=9 style="HEIGHT: 24px; WIDTH: 127px"></TD>
<TD width="20%"></TD>
<TD width="33%"><INPUT onclick="return validateForm()" name=submit1 type=submit
value="Submit Request" tabIndex=8></TD>
</TR></TABLE></FORM>

<HR size=3 color=white>
<TABLE align=center border=0 cellPadding=1 cellSpacing=1 width="100%">

<TR>
<TD width="30%" height="23">
<center>

</center></TD>

<TD width="30%"></TD>

<TD width="30%"><center>

</center></TD>
</TR>

<TR>
<TD width="30%"><center>
Main Menu</center></TD>
<TD width="30%">
</TD>

<TD width="30%"><center>
Logout </center></TD>

</TR> </TABLE>

</BODY>
</HTML>

2. Request for an Article of a Journal Verification

```
<%@ Language=VBScript %>
<%
'Current Session Journal Details
Dim strJournalTitle
Dim strArticleAuthor
Dim strArticleTitle
Dim strPublishDate
Dim strVol
Dim strIssue
Dim strArticlePage
Dim strType

strJournalTitle = Request.Form ("JournalTitle")
strArticleAuthor = Request.Form ("ArticleAuthor")
strArticleTitle = Request.Form ("ArticleTitle")
strPublishDate = Request.Form ("PublishDate")
strVol = Request.Form ("Vol")
strIssue = Request.Form ("Issue")
strArticlePage = Request.Form ("ArticlePage")
strType = Request.Form ("Type")

Session("Journal_Title")= strJournalTitle
Session("Journal_ArticleAuthor")= strArticleAuthor
Session("Journal_ArticleTitle") = strArticleTitle
Session("Journal_PublishDate") = strPublishDate
Session("Journal_Vol")= strVol
Session("Journal_Issue")= strIssue
Session("Journal_ArticlePage")= strArticlePage
Session("Journal_Type")= strType
%>

<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</HEAD>

<BODY topmargin=0 background="Pic\Background_Reg.GIF">
<IMG hspace=50 src="pic/journal.gif"><HR size=3 color=white>

<FORM METHOD="post" ACTION="ConfJournal.asp">
<TABLE background="Pic\Background.GIF" align=center border=1 cellPadding=0
cellSpacing=1 id=TABLE1 width="80%" style="HEIGHT: 238px; WIDTH: 608px">

<TR>
<TD width="20%" valign=top><FONT size=4>Name</FONT></TD>
<TD width="70%"><FONT face=arial size=3><I><B>
<LABEL style="HEIGHT: 22px"><%=Session("CurrentUserName")%></LABEL>
</I></B></FONT></TD></TR>
```



```
<TR>
  <TD valign=top><FONT size=4 >User ID</FONT></TD>
  <TD><FONT face=arial size=3><I><B>
    <LABEL style="HEIGHT: 22px"><%=Session("CurrentUserID")%></LABEL>
  </I></B></FONT></TD></TR>

<TR>
  <TD valign=top><FONT size=4 >eMail</FONT></TD>
  <TD><FONT face=arial size=3><I><B>
    <LABEL style="HEIGHT: 22px"><%=Session("CurrentUserEmail")%></LABEL>
  </I></B></FONT></TD></TR>

  <TR>
  <TD valign=top>
    <FONT size=4 >Journal Title</FONT>
  </TD>
  <TD>
    <FONT face=arial size=3><I><B>
    <LABEL style="HEIGHT: 22px"><%=Request.Form ("JournalTitle")%></LABEL>
  </I></B></FONT>
</TD> </TR>

<TR>
  <TD valign=top> <FONT size=4 >Article Author</FONT>
  </TD>
  <TD>
    <FONT face=arial size=3><I><B>
    <LABEL style="HEIGHT: 22px"><%=Request.Form ("ArticleAuthor")%></LABEL>
  </I></B></FONT></TD></TR>

<TR>
  <TD valign=top>
    <FONT size=4 >Article Title</FONT>
  </TD>
  <TD><FONT face=arial size=3><I><B>
    <LABEL style="HEIGHT: 22px"><%=Request.Form ("ArticleTitle")%></LABEL>
  </I></B></FONT></TD></TR>

<TR>
  <TD valign=top><FONT size=4 >Publish Date</FONT>
  </TD>
  <TD><FONT face=arial size=3><I><B>
    <LABEL style="HEIGHT: 22px"><%=Request.Form ("PublishDate")%></LABEL>
  </I></B></FONT></TD>
</TR>

<TR>
  <TD valign=top>
    <FONT size=4 >Volume</FONT></TD>
  <TD><FONT face=arial size=3><I><B>
    <LABEL style="HEIGHT: 22px"><%=Request.Form ("Vol")%></LABEL>
```



```
</I></B></FONT></TD></TR>
<TR>
<TD valign=top><FONT size=4 >Issue Number</FONT></TD>
<TD><FONT face=arial size=3><I><B>
<LABEL style="HEIGHT: 22px"><%=Request.Form ("Issue")%></LABEL>
</I></B></FONT>
</TD></TR>

<TR><TD valign=top><FONT size=4 >Pages of Articles</FONT></TD>
<TD><FONT face=arial size=3><I><B>
<LABEL style="HEIGHT: 22px"><%=Request.Form ("ArticlePage")%></LABEL>
</I></B></FONT> </TD></TR>

<TR> <TD valign=top><FONT size=4 >Type</FONT> </TD>
<TD><FONT face=arial size=3><I><B>
<LABEL style="HEIGHT: 2px"><%=Request.Form ("Type")%></LABEL>
</I></B></FONT> </TD></TR></TABLE>

<TABLE align=center border=0 cellpadding=1 cellspacing=1 width="75%">

<TR> <TD width="40%"><A href="/edds/RequestMenu.asp">
<IMG align=bottom border=0 height=35 src="ico/Reset.gif" width=35 alt="Cancel current
request"></A>

<A href="/edds/RequestMenu.asp">
<font color=maroon size=4>Cancel Request</font></A></TD>

<TD width="33%"></TD> <TD>
<INPUT id=submit1 name=submit1 style="FONT-SIZE: small" type=submit value="Confirm
Request"></TD></TR></TABLE><P></P>
</FORM>
</BODY>
</HTML>

if not objRS.EOF then
    strProcess= "Yes"
End If
%>
<HTML>
<HEAD>

<script language="javascript">

function validateForm() {
var Source=document.getElementById("formSource").value
var email=document.getElementById("emailSource").value
```


3. Request Processing Module

```
<%@ Language=VBScript %>
<!--#include file="dbaseConn.inc"-->
<%
    Dim strCurrentUser
    strCurrentUser = Session("CurrentUserID")
    If strCurrentUser <> "admin" then
%>
<P align=center><Font size=5 color=red><B>You are restricted to view this
page!</B></FONT></P>
<% Else %>
<%
```

```
    Dim sql
```

```
    Dim strProcess
    strProcess="No"
```

```
    Dim category
    category = Request.Form("category")
    category = cInt(category)
    Session("Category")=category
```

```
IF category = 1 Then
```

```
    sql = "SELECT * FROM Journal WHERE Req_Status = 0"
```

```
Else
```

```
    sql = "SELECT * FROM BookChap WHERE Req_Status = 0"
```

```
End IF
```

```
    Dim objRS
```

```
    Set objRs = Server.CreateObject("ADODB.Recordset")
```

```
    objRs.Open sql, objConn, 3, 3, 1
```

```
if not objRS.EOF then
```

```
strProcess="Yes"
```

```
End If
```

```
%>
```

```
<HTML>
```

```
<HEAD>
```

```
<script language="JavaScript">
```

```
function validateForm() {
```

```
var Source=document.processForm.Source.value
```

```
var eMail=document.processForm.eMailSource.value
```



```
if(Source==""){ alert("Please specify the source for this document")
    document.processForm.Source.focus()
    return false}
if(eMail==""){ alert("Please type in the source Email Address!")
    document.processForm.eMailSource.focus()
    return false}
if(eMail.indexOf('@', 0) == -1 || eMail.indexOf('.', 0) == -1)
{
    alert("Please enter a valid email address!");
    document.processForm.eMailSource.focus()
    return false}
}</script>

<META NAME="GENERATOR" Content="Microsoft Visual Studio 6.0"></HEAD>

<BODY background="Pic\login_back.gif" topmargin=0>

<% If category = 1 then %>
<Center></Center><HR size=3 color=gray>
<% Else %>
<Center></Center><HR size=3 color=gray>
<% End If %>

<FORM Method=post action="ProcessReq.asp" name="processForm">
<%If strProcess = "No" Then %>
<P align=center><FONT color=red size=6 face=Verdana>No request available</FONT></P>

<% Else %>
<% IF category = 1 Then %>
<% Session("ReqID")=objRs("Req_ID")%>
<TABLE align=center border=1 cellPadding=1 cellSpacing=1 id=TABLE1 width="80%"
style="HEIGHT: 204px; WIDTH: 80%">
<TR>
    <TD><FONT size=4>Name</FONT></TD>
    <TD><INPUT name=text1 readOnly
        style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("Name")%>"></TD></TR>
<TR><TD><FONT size=4>User ID</FONT></TD>
    <TD><INPUT name=text2 readOnly value="<%=objRs("UserID")%>"></TD></TR>
<TR><TD><FONT size=4>eMail</FONT></TD>
    <TD><INPUT name=text1 readOnly
        style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("eMail")%>"></TD></TR>

    <TR> <TD> <FONT size=4>Journal Title</FONT> </TD>
    <TD> <FONT size=4><INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("JournalTitle")%>" name=text3></FONT>
    </TD> </TR>

    <TR> <TD> <FONT size=4>Article Author</FONT> </TD>
    <TD> <FONT size=4> <INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("ArticleAuthor")%>" name=text4> </FONT> </TD> </TR>
```



```

<TR> <TD> <FONT size=4>Article Title</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("ArticleTitle")%>" name=text5></FONT> </TD>.. </TR>

<TR> <TD> <FONT size=4>Publish Date</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly
style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("PublishDate")%>"
name=text6></FONT> </TD> </TR>

<TR> <TD> <FONT size=4>Volume</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("Vol")%>" name=text7></FONT> </TD>
</TR>

<TR> <TD> <FONT size=4>Issue Number</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("Issue")%>" name=text8></FONT>
</TD>
</TR>

<TR> <TD> <FONT size=4>Pages of Articles</FONT> </TD>
<TD><FONT size=4><INPUT
readOnly style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("ArticlePage")%>"
name=text9></FONT> </TD> </TR>
<TR> <TD> <FONT size=4>Source</FONT> </TD>
<TD> <FONT size=4>
<INPUT type="text" style="HEIGHT: 22px; WIDTH: 320px" name=Source></FONT>
</TD></TR>
<TR> <TD> <FONT size=4>Source email</FONT> </TD>
<TD> <FONT size=4> <INPUT type="text" style="HEIGHT: 22px; WIDTH: 280px"
name=eMailSource></FONT></TD></TR> </TABLE>

<% else %>
<% Session("ReqID")=objRS("Req_ID")%>
<TABLE align=center border=1 cellPadding=1 cellSpacing=1 id=TABLE1 width="80%"
style="HEIGHT: 204px; WIDTH: 80%">

<TR> <TD><FONT size=4>Name</FONT></TD>
<TD><INPUT name=text1 readOnly
style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("Name")%>"></TD></TR>
<TR> <TD><FONT size=4>User ID</FONT></TD>
<TD><INPUT name=text2 readOnly value="<%=objRs("UserID")%>"></TD></TR>

<TR><TD><FONT size=4>eMail</FONT></TD>
<TD><INPUT name=text2 readOnly value="<%=objRs("eMail")%>"></TD></TR>

<TR> <TD> <FONT size=4>Chapter Author</FONT></TD>
<TD> <FONT size=4><INPUT readOnly
style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("ChapAuth")%>"
name=text10></FONT> </TD> </TR>

```



```
<TR> <TD> <FONT size=4>Chapter Title</FONT> </TD>
<TD> <FONT size=4> <INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("ChapTitle")%">" name=text11> </FONT> </TD> </TR>

<TR> <TD> <FONT size=4>Title of Book</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly
style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("BookTitle")%">" name=text12>
</FONT> </TD> </TR>

<TR> <TD> <FONT size=4>Pages in Book</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly
style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("BookPages")%">"
name=text13></FONT> </TD> </TR>

<TR> <TD> <FONT size=4>Publisher</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("Publisher")%">" name=text14></FONT> </TD> </TR>

<TR> <TD> <FONT size=4>Date</FONT> </TD>
<TD> <FONT size=4><INPUT
readOnly style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("PublishDate")%">"
id=text15 name=text15></FONT> </TD> </TR>

<TR> <TD> <FONT size=4>Place Publish</FONT> </TD>
<TD> <FONT size=4><INPUT readOnly style="HEIGHT: 22px; WIDTH: 280px"
value="<%=objRs("PublishPlace")%">" id=text16 name=text16></FONT>
</TD> </TR>

<TR> <TD> <FONT size=4>ISBN</FONT> </TD>
<TD> <FONT size=4><INPUT
readOnly style="HEIGHT: 22px; WIDTH: 280px" value="<%=objRs("ISBN")%">"
id=text3 name=text3></FONT>
</TD> </TR>

<TR> <TD> <FONT size=4>Source</FONT> </TD>
<TD> <FONT size=4>
<INPUT type="text" style="HEIGHT: 22px; WIDTH: 320px" name=Source></FONT>
</TD></TR>

<TR> <TD> <FONT size=4>Source email</FONT> </TD>
<TD><FONT size=4><INPUT type="text" style="HEIGHT: 22px; WIDTH: 280px"
name=eMailSource></FONT></TD></TR>

</TABLE>
<P>
<% End If %>

<TABLE cellSpacing=1 cellPadding=1 width="339" align=center border=0 style="HEIGHT:
35px; WIDTH: 339px">
<TR height=20></TR>
<TR>
```



```
<TD align=left><INPUT type=reset value=Reset id=reset1 name=reset1 style="FONT-SIZE:
small; HEIGHT: 31px; WIDTH: 100px" size=33></TD>
<TD></TD>
<TD align=right><INPUT type=submit onclick="return validateForm()" value=Process
id=submit1 name=submit1 style="FONT-SIZE: small; HEIGHT: 31px; WIDTH: 100px"
size=33></TD></TR></TABLE>
</FORM>

<% End If %>
<%
        objRs.Close
        Set objRS=Nothing
        objConn.Close
        Set objConn= Nothing
%>
<HR size=3 color=gray>
<TABLE align=center border=0 cellPadding=0 cellSpacing=0 width="100%" style="HEIGHT:
105px; WIDTH: 760px"><!-- First Row icon -->
<TR>
<TD width="30%" height="10"><center><A href="/edds/AdminMenu.asp">
<IMG align=bottom border=0 height=35 src="ico/Main.gif" width=35 alt="Main Menu">
</A></center></TD>

<TD width="30%" height="10"><center><A href="/edds/ProcessMenu.asp">
<IMG align=bottom border=0 height=35 src="ico/process.gif" width=35 alt="Process Request
Menu">
</A></center> </TD>

<TD width="30%" height="10"><center><A href="/edds/Logout.asp">
<IMG align=bottom border=0 height=35 src="ico/Logout.gif" width=35 alt="Logout">
</A></center> </TD> </TR>
<!-- First Row links -->
<TR>
<TD width="30%" height="10"><center><A href="/edds/AdminMenu.asp">
<FONT color=blue face=serif size=4>Main Menu</FONT>
</A></center></TD>

<TD width="30%" height="10">
<center><A href="/edds/ProcessMenu.asp">
<FONT color=blue face=serif size=4>Process Request Menu</FONT></A></center></TD>

<TD width="30%" height="10"> <center><A href="/edds/Logout.asp">
<FONT color=blue face=serif size=4>Logout</FONT> </A> </center> </TD> </TR>
<tr height="25"></tr>
</TABLE>

<%End If%>
</BODY>
</HTML>
```


References

[1]	Carolyn Weaver, "Electronic document delivery", in Encyclopedia of library and information science, Volume 40, Supplement 5, ed. Allen Kent (New York: Marcel Dekker inc., 1986), p.43
[2]	William Tuck, Document delivery and high-speed networks, Online Information 92, 15th International Online Information Meeting Proceedings London 8-10 December 1992. (Oxford: Learned Information, 1992), p.7
[3]	Hans Roes and Joris Dykstra, January 1994 "Aranda: the next generation electronic document delivery system" online at: http://cwis.kub.nl/~del/users/roes/articles/article.htm (Accessed on 27 June 2001)
[4]	University of Nevada, Reno Libraries online at: http://www.library.unr.edu/~lib/dccdel/default.htm (Accessed on 20 July 2001)
[5]	University of Illinois at Chicago Libraries online at: http://www.library.uic.edu/~lib/dccdel/default.htm (Accessed on 20 July 2001)
[6]	University of Illinois at Chicago Libraries online at: http://www.library.uic.edu/~lib/dccdel/default.htm (Accessed on 20 July 2001)
[7]	Arizona State University Libraries online at: http://www.asu.edu/lib/ (Accessed on 20 July 2001)
[8]	RMIT University Library online at: http://www.lib.rmit.edu.au/ (Accessed on 20 July 2001)
[9]	Monash University Library online at: http://www.library.monash.edu.au/dccdel/ (Accessed on 20 July 2001)
[10]	SHPL online at: http://www.lib.monash.edu.au/dccdel/ (Accessed on 20 July 2001)
[11]	Medical University of South Carolina (MUSC) online at: http://www.library.musc.edu/home.html (Accessed on 30 July 2001)
[12]	National Library of Australia Loanable Doc online at: http://www.nla.gov.au/doc/loanable.html (Accessed on 30 July 2001)
[13]	The Clement C. Maxwell Library online at: http://www.bridgew.edu/depts/maxwell/library.htm (Accessed on 30 July 2001)

References

[1]	Carolyn Weaver, "Electronic document delivery", in Encyclopedia of library and information science, Volume 40, Supplement 5; ed. Allen Kent. (New York: Marcel Dekker inc., 1986), p.43
[2]	William Tuck, Document delivery and high-speed networks, Online Information 92: 16th International Online Information Meeting Proceedings London 8-10 December 1992. (Oxford: Learned Information, 1992), p.7.
[3]	Hans Roes and Joost Dijkstra, January 1994 "Ariadne: the next generation electronic document delivery systems" online at: http://cwis.kub.nl/~dbi/users/roes/articles/ariadne.htm (Accessed on 28 June 2001)
[4]	University of Nevada, Reno Libraries online at: www.library.unr.edu/depts./docdel/default.htm (Accessed on 20 July 2001)
[5]	Syracuse University Library online at: http://libwww.syr.edu/ (Accessed on 20 July 2001)
[6]	Stanford University Library online at: http://www-sul.Stanford.edu (Accessed on 20 July 2001)
[7]	Arizona State University Libraries online at: http://www.asu.edu/lib (Accessed on 20 July 2001)
[8]	RMIT University Library online at: http://www.lib.rmit.edu.au/dds (Accessed on 20 July 2001)
[9]	Monash University Library online at: http://www.lib.monash.edu.au/docdel (Accessed on 20 July 2001)
[10]	SHPL online at: http://www.lib.monash.edu.au/docdel (Accessed on 30 July 2001)
[11]	Medical University of South Carolina (MUSC) online at: http://www.library.musc.edu/home.html (Accessed on 30 July 2001)
[12]	National Library of Australia Loansome Doc online at: http://www.nla.gov.au/dss/loansome.html (Accessed on 30 July 2001)
[13]	The Clement C. Maxwell Library online at: http://www.bridgew.edu/depts/maxwell/library.htm (Accessed on 30 July 2001)

[14]	Ronald LeRoi Burback, July 1997 "Methodology" online at : http://www-db.stanford.edu/~burback/water_sluice/sluice6.2.25.97/ws/node300..html (Accessed on 27 July 2001)
[15]	Ronald LeRoi Burback, July 1997 "Four Fundamental Phases" online at : http://www-db.stanford.edu/~burback/water_sluice/sluice6.2.25.97/ws/node7.html (Accessed on 27 July 2001)
[16]	Ronald LeRoi Burback, July 1997 "Classical Waterfall Methodology" online at : http://www-db.stanford.edu/~burback/water_sluice/sluice6.2.25.97/ws/node50.html (Accessed on 27 July 2001)
[17]	Ronald LeRoi Burback, July 1997 "Classical Waterfall Methodology, Why it works" online at : http://www-db.stanford.edu/~burback/water_sluice/sluice6.2.25.97/ws/node51.html (Accessed on 27 July 2001)
[18]	Pfleeger S.L., 1998 "Software Engineering, Theory and Practice". Upper Saddle River, NJ: Prentice-Hall
[19]	Ronald LeRoi Burback, July 1997 "Four Fundamental Phases, The Analysis Phase" online at : http://www-db.stanford.edu/~burback/water_sluice/sluice6.2.25.97/ws/node8.html (Accessed on 27 July 2001)
[20]	Ronald LeRoi Burback, July 1997 "Four Fundamental Phases, The Design Phase" online at : http://www-db.stanford.edu/~burback/water_sluice/sluice6.2.25.97/ws/node15.html (Accessed on 27 July 2001)
[21]	Ronald LeRoi Burback, July 1997 "Four Fundamental Phases, The Implementation Phase" online at : http://www-db.stanford.edu/~burback/water_sluice/sluice6.2.25.97/ws/node21.html (Accessed on 27 July 2001)

- | | |
|------|---|
| [22] | Ronald LeRoi Burbach, July 1997 “ Four Fundamental Phases, The Testing Phase”
online at :
http://www-db.stanford.edu/~burbach/water_sluice/slucose6.2.25.97/ws/node23.html
(Accessed on 27 July 2001) |
| [23] | Mitchell S. & Atkinson J., January 2000 “SAMS Teach Yourself Active Server Pages
3.0”. Indiana: Sams Publishing. |
| [24] | Tittel E., Pitts N. & Valentine C., 2000 “HTML 4 For Dummies”(3 rd Ed.). Foster City:
IDG Books Worldwide Inc. |
| [25] | Webopedia Definition and Links “WWW” online at :
http://webopedia.internet.com/TERM/W/World_Wide_Web.html
(Accessed on 2 August 2001) |
| [26] | Webopedia Definition and Links “FTP” online at :
http://webopedia.internet.com/TERM/F/File_Transfer_Protocol.html
(Accessed on 2 August 2001) |
| [27] | Whitten J.L., Bentley L.D. & Dittman K.C., (2000).”System Analysis And Design
Methods” (5 th Ed.). New York: McGraw-Hill. |
| [28] | Walther S. & Levine J., April 2000 “SAMS Teach Yourself E-Commerce Programming
with ASP”. Indiana: Sams Publishing. |