Digital Library of Theses

Administration Side

CHEW SOO NOAN

Under Supervision of

DR. SELVANATHAN NARAINASAMY

This project is submitted to the

Faculty of Computer Science and Information Technology

University of Malaya

In

Partial fulfillment of the requirement for the Degree of

Bachelor of Information Technology

Session 2000/2001

Abstract

Currently, the faculty promoted a plan to rethink the way graduate students present their theses. Traditionally, the theses were produced in paper form and storing in library. The access of student work through this method still facing a lot of problems including less timely public access to users, shelf space required for storage, no maintenance, resources are not fully used and cost of preparing theses are high. Thus, Digital Library of Theses is proposed to eliminate all the shortcomings mentioned above.

Digital Library of Theses (administration side) is a web-based theses administration system for faculty that can cater about 3 years theses and other records. The system contains three main modules or subsystem, namely theses publication, maintenance and user access and queries. There will be two administrators, super administrator and thesis administrator managing the system with the function like add and edit faculty and department, generate reports, add and edit supervisor and etcetera to support the client side of system. The entire system is based on three-tier client/server architecture. All communications of system are done using Intranet through TCP/IP protocol.

With the user-friendly interfaces, Digital Library of Theses ease administrator for remote managing and system maintenance. In addition, it helps administrator to manage theses time-effectively with less manpower. The completion of this web-based digital library theses system allows lecturer and student to access it within the local area network of faculty.

ACKNOWLEDGEMENT

The author wishes to express his appreciation and thanks to the following persons for their helps and guidance.

Heartfelt gratitude to Dr. Selvanathan, the project supervisor for his precious advice, guidance, encouragement, patience and constructive suggestions. Special thanks to Dr. Roziati, the project Moderators for her comments and suggestions. Not forget to thanks the lecturers of Faculty of Computer Science and Information Technology, who had help to give us the precious advice for this project. A special notes of thanks also given to Mrs. Khor for helping us a lot in the laboratory.

Utmost gratitude to the members of this project group, Miss Chong Fong Fong and Miss Chang Chai Ling for being cooperation with me during the project building. A thousand of thanks to Mr. Jimmy Tan, Kenny Teoh and other fellow coursemates for giving valuable idea and comments for the project. A special note of thanks also gives to my housemates for helping me preparing the thesis. To my family, I am grateful for their unfailing support and encouragement throughout the years.

TABLE OF CONTENT

ABSTRACT I

ACKNOWLEDGEMENT II

TABLE OF CONTENT III

LIST OF FIGURES VIII

LIST OF TABLES IX

CHAPTER 1 INTRODUCTION **1**

1.1 PROJECT OVERVIEW 1

1.2 PROJECT OBJECTIVE 2

1.3 PROJECT SCOPE 3

1.3.1 Theses Publication 3

1.3.2 User Access & Queries 3

1.3.3 Maintenance 4

1.4 PROJECT DEVELOPMENT STRATEGY 5

1.5 PROJECT SCHEDULE 7

1.6 RESEARCH CONDUCTED 9

1.7 REPORT ORGANIZATION 10

CHAPTER 2 LIRTERATURE REVIEW **11**

2.1 TRADITIONAL THESES SYSTEM 11

2.1.1 Introduction 11

2.1.2 Weakness and Drawbacks of Traditional Theses System 12

2.2 ELECTRONIC THESES AND DISSERTATION 13

2.2.1 Introduction 13

2.2.2 History of ETDs 14

2.3 INITIATIVES 15

2.3.1 Student Interest 15

2.3.2 Other Related Efforts 16

2.3.3 VT ETD Initiatives 17

2.4 STATISTICS 18

2.4.1 Statistics 18

2.5 CASE STUDIES 20

2.5.1 University of Microfilm International (UMI) 20

2.5.2 West Virginia University (WVU) 22

2.5.3 NDLTD 23

2.5.4 Internet Library Administration System (NetLibrary) 24

2.5.5 Electronic Journal System (EJGU) 25

2.5.6 Features Highlighting 25

2.6 BENEFITS OF ETD 27

2.7 OTHER ISSUES 27

2.7.1 Archiving 27

2.7.2 Standard Electronic Formats 29

2.8 CONCLUSION 32

CHAPTER 3 SYSTEM ANALYSIS 33

3.1 INTRODUCTION 33

3.2 REQUIREMENTS DEFINING'S TECHNIQUES 33

3.3 SYSTEM MODEL 34

3.4 FUNCTIONAL REQUIREMENTS 35

3.4.1 Theses Publication 35

3.4.2 Maintenance 36

3.4.3 User Access & Queries 36

3.5 NON-FUNCTIONAL REQUIREMENTS 36

3.5.1 Graphical User Interface (GUI) 36

3.5.2 Reliability 37

3.5.3 Robustness 37

3.5.4 Maintainability 37

3.5.5 Correctness 37

3.6 PLATFORM AND WEB SERVER CONSIDERATION 38

3.6.1 Windows Microsoft NT Server 38

3.6.2 Microsoft Internet Information Server 3.0 38

3.6.3 Unix Operating System 39

3.6.4 Preference of Operating System and Web Server 41
3.7 PROGRAMMING LANGUAGES AND COMPONENTS CONSIDERATION 41
3.7.1 Component Object Model (COM) 42
3.7.2 Distributed Common Object Model (DCOM) 46
3.7.3 Common Object Request Broker Architecture (CORBA) 49
3.7.4 COM/DCOM vs. CORBA 53
3.7.5 ActiveX 54
3.7.6 Java 58
3.7.7 ActiveX vs. Java 60
3.7.8 Active Server Pages (ASP) 62
3.7.9 Common Gateway Interface (CGI) 62
3.7.10 ASP vs. CGI 63
3.7.11 Other consideration 64
3.7.11.1 Visual Basic 64
3.7.11.2 VBScript 65
3.7.12 Conclusion 65
3.8 HARDWARE AND SOFTWARE REQUIREMENTS 67
3.8.1 Server side 67
3.8.2 Client side 68
CHAPTER 4 SYSTEM DESIGN 69
4.1 INTRODUCTION 69
4.2 SYSTEM FUNCTIONALITY DESIGN 69
4.2.1 Client/Server 69
4.2.2 Digital Library of Theses 70
4.2.3 Overall Data Flow Diagram 72
4.2.4 System Structure Chart 75
4.3 DATABASE DESIGN 76
4.3.1 Normalization 76
4.3.2 Data Dictionary 76
4.4 USER INTERFACE DESIGN 80
4.3.1 Screen Design 81

4.3.2 Navigation Bar 82

4.3.3 Design of Menu 84

CHAPTER 5 SYSTEM IMPLEMENTATION 89

5.1 INTRODUCTION 89

5.2 CODING 89

5.2.1 Coding Methodology 89

5.2.2 Administration Side Coding 90

5.3 TESTING 93

5.3.1 Unit Testing 94

5.3.2 Integration Testing 95

5.3.3 System Testing 96

5.4 SOFTWARE TOOLS 96

5.4.1 Development Tools 96

5.4.2 Use Interfaces Design Tools 97

5.4.3 Documentation Tools 97

CHAPTER 6 EVALUATION AND CONCLUSION 98

6.1 PROBLEM AND SOLUTIONS 98

6.1.1 Lack of knowledge and experience in web-based Programming 98

6.1.2 Lack of knowledge and experience in interface designing 98

6.1.3 Lack of knowledge and experience in database server 99

6.2 SYSTEM STRENGTHS 99

6.2.1 Friendly User Interface 99

6.2.2 Time-effectively with Less Manpower Theses System 99

6.2.3 Error Controlling 99

6.3 SYSTEM LIMITATION 100

6.3.1 Browser Limitation 100

6.3.2 No Context Sensitive Help 100

6.3.3 No Encryption for password 100

6.3.4 Unsupport dual Lingual 100

6.4 FUTURE ENHANCEMENT 101

6.4.1 Security 101

6.4.2 Animated User Interface 101

6.4.3 Online Theses System 101

6.4.4 Subscription and Online Payment 101

6.4.5 Editorial Board 102

6.4.6 Mail Server 102

6.4.7 Secure Socket Layer 102

6.4.8 Dual Lingual Support 102

6.5 CONCLUSION 103

BIBIOGRAPHY 104

APPENDIX: User Manual 107

LIST OF FIGURES

Figure 1.1 Prototyping Model 5

Figure 1.2 Project Schedule 8

Figure 2.1 Traditional Theses System 11

Figure 3.1 Digital Library of Theses system model 34

Figure 3.2 COM components in different processes 47

Figure 3.3 DCOM: COM components on different machines 47

Figure 3.4 Integrating HTML and distributed computing 49

Figure 3.5 CORBA ORB Architecture 51

Figure 3.6 Four ways of ActiveX 57

Figure 3.7 How Java Works 58

Figure 3.8 Java API and visual machine 59

Figure 4.1 Digital Library System Three-Tier Client Server Architecture 71

Figure 4.2 Super Administrator's data flow diagram 73

Figure 4.3 Theses Administrator's data flow diagram 74

Figure 4.4 System Structure Chart 75

Figure 4.5 Interface Design 81

Figure 4.6 Navigation Bar 82

Figure 4.7 Super Administrator Menu 85

Figure 4.8 Theses Administrator Menu 87

LIST OF TABLES

Table 2.1 Access Statistics 18

Table 2.2 Accesses by Internet Domain 19

Table 2.3 Download by country for 1996 and 1997 showing increases 19

Table 2.4 Administration features comparison for e-theses system, e-journal system and e-library system 26

Table 2.5 File types and format 30

Table 3.1 Comparison between major features for Java and ActiveX 60

Table 3.2 Comparison between major options for CGI and ASP 63

Table 3.3 Server Software Requirements 67

Table 4.1 Table SuperAdmin 76

Table 4.2 Table ThesisAdmin 77

Table 4.3 Table Faculty 77

Table 4.4 Table Department 78

Table 4.5 Table Supervisor 78

Table 4.6 Table Login Student 79

Table 4.7 Table Announcement 79

Table 4.8 Super Administrator Menu 86

Table 4.9 Theses Administrator Menu 88

Table 5.1 Administrator-side maintenance coding 90

Table 5.2 Administrator-side thesis publication (super administrator) coding 91

Table 5.3 Administrator-side thesis publication (thesis administrator) coding 92

Table 5.4 Administrator-side user access and queries coding 93

Table 5.5 System development tools 96

Table 5.6 Description of graphical tools 97

Table 5.7 Description of documentation tools 97

Chapter One

Introduction

CHAPTER 1 INTRODUCTION

1.1 PROJECT OVERVIEW

Digital Library of Theses is transforming graduate education. Requiring students to author electronic theses (or offering it as an option) introduces graduate students, faculty, and libraries to electronic publishing. Digital Libraries of Theses enable a university to widely distribute the intellectual products of its graduates and to introduce its students to the Knowledge Age. Graduate students are defending proposals and dissertations online, and are able to complete their dissertations without printing a word.

The Digital Library of Theses for administration is a web-based e-thesis administration system for University of Malaya generally and for Faculty of Computer Science and Information Technology (FSKTM) specifically that enable administrator to maintain and access the theses in an efficient and convenient manner.

The entire system is based on three-tier client/server architecture that consists of the client tier, a middle tier (consist of business rules that reside on web server) and the data source (database server). Administrators can access the system from web server using TCP/IP protocol (as a transmission medium for the above 3-tier to communicate) in Local Area Network (LAN). The object or component (middle-tier) that reside on web server will process all the data request from clients before the data is entered into or retrieved from database server. No direct connection is established between clients and database server.

The entire system and its business rules reside on the web server and the entire data is kept on the database server (database source). Client serves as a tool to use and access the system from remote location.

The entire digital thesis system enables undergraduates and lecturers to read the latest theses within the Intranet in FSKTM faculty. It makes the system maintenance possible and easily.

1.2 PROJECT OBJECTIVES

The main objective of the project is to develop a web-based thesis administration system for Faculty of Computer Science and Information Technology (FSKTM) that enable the administrator to use and maintain the thesis system from any location within the faculty. Besides the main objective, there is other objectives listed below:

- Development of faculty and students as electronic scholars.
- Be easy for the administrator or user to use without extensive training in computer techniques or extensive changes in library principles.
- Provide an easy to use and user-friendly graphical user interfaces.
- Provide a simple method of updating computer records.
- Able to cater a large amount of bibliographic data through the use of back-end database server.
- Enable the process of remote maintenance through the Intranet of faculty.
- To ensure authorized access to the administration function of the system.
- To reduce the manpower in clerical function of library system.

1.3 PROJECT SCOPE

The project will cover three major modules, namely theses publication, maintenance and user access and queries.

1.3.1 Theses Publication

In order to manage the thesis publication for student, its presence in digital library must be recorded and also other information related. This process is known as theses publication which contains three level of administrator including super administrator, theses administrator and assistance administrator. Super administrator has the highest priority in this module. The super administrator can grant permission to any theses administration for managing the department creation and adding any assistance administrator to assist her or him. Assistance administrator is managing the welfare of the author/ student regarding the thesis. Through this module, administrators are able to add, edit, delete, search information of author and their metadata (including their personal details, thesis title, abstract etc.).

1.3.2 User Access and Queries

A user access and queries module enable authorized administrator to login to the system and logout for controlling the security of the system. This module also enables the administrator the change their personal such as email, login name and password. Any queries or feedback will be sent back to user through email.

1.3.3 Maintenance

Maintenance allows authorized administrator to place an online-announcement in web page. Besides, this module also enables administrator to manage the preservation of theses by using archive and backup function. Administrator may write the previous year theses to CD-ROMs, which also preserved as a backup. On the other hand, administrator might assign a code number to CD-ROM that store the previous year theses in database for further reference and ease the search function.

1.4 PROJECT DEVELOPMENT STRATEGY

Prototyping development is an idea of developing an initial implementation, exposing this to user comment and refining this through many versions until an adequate system has been developed. Rather than have separate specification, development and validation activities, these are carried out concurrently with rapid feed back across these activities. It is a working model of the system built to learn about its true requirements. The sequence of the events for the prototyping paradigm is shown in figure 1.1.

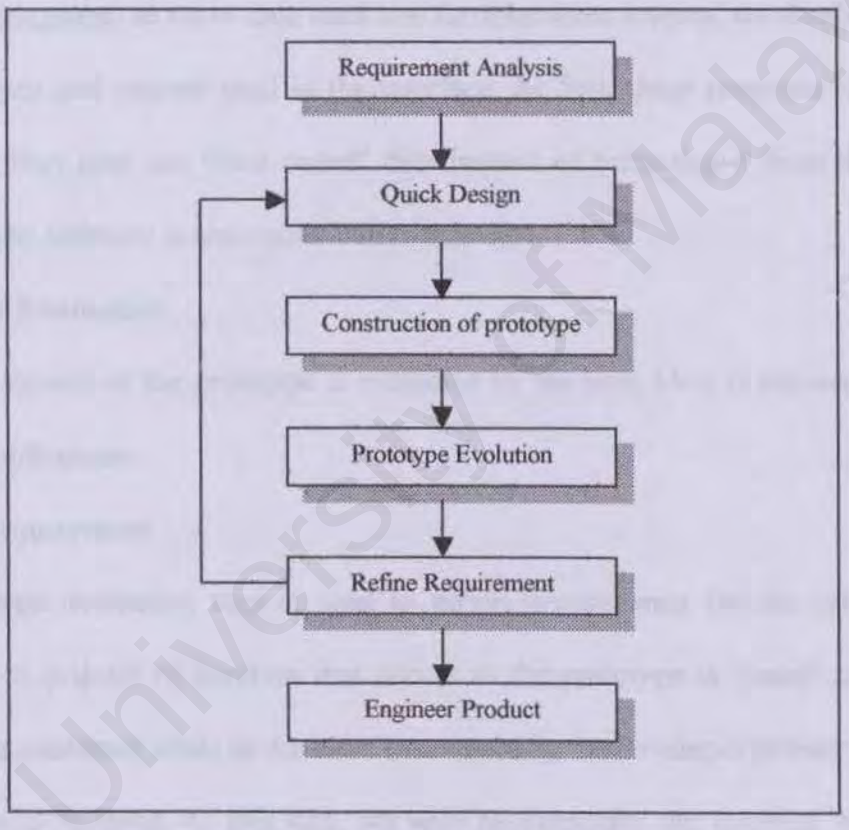


Figure 1.1 Prototyping Model

• Requirement Analysis

Prototyping begins with requirements gathering and analysis. In this step, developer and user meet and define the overall objectives for the software, identify whatever requirements are known, and outline areas where further definition is mandatory.

• Quick Design

After the requirement analysis phase is done, a "quick design" then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the user (e.g., input. approaches and output formats). The quick design leads to the construction of a prototype.

• Construction of Prototype

Prototype software is created and it will consist of programs (usually written in a fourth generation language) to move data back and forth between screens, the database, reports, and the inputs and outputs used in the interface. At first, these programs may do little processing; they may use "hard coded" data instead of extracting it from the database. The prototype software is also tested and redefined.

• Prototype Evaluation

The tested version of the prototype is evaluated by the user. User is allowed to test and suggest modifications.

• Refine Requirement

The prototype evaluation step is used to refine requirements for the software to be developed. A process of iteration that occurs as the prototype is "tuned" to satisfy the needs of the customer while at the same time enabling the developer to better understand what needs to be done. At this step, not only requirements are justified, but also new requirements are added when possible.

• Engineer Product

Step from quick design to redefine requirement is repeated iteratively until all requirements are formalized or until the prototype has evolved into a production system.

The prototyping method used for this project is evolutionary prototype. Evolutionary Prototype is designed to be adapted for permanent use after the ideas are clarified and must be built using the programming tools that will be used for the final system. Prototyping model is selected for the development of this project because:

- Prototyping model can be built quickly.
- The prototype provides the user with a tangible means of comprehending and evaluating the proposed system and elicits more meaningful feedback.
- Misunderstanding of the requirement can be identified in the early stage.
- Missing functions can also be identified.
- Difficult-to-use or confusing user requirements may be identified and redefined.
- The prototype provides a common baseline and frame of reference. Developers and users can communicate better.
- The prototype helps ensure system performs adequately before widespread release.

1.6 PROJECT SCHEDULE

Time management is very important for project developer. It serves as a vehicle in determining how deep the problems has been analyze, how comprehensive should a solution be designed to overcome the problem, how complete is a source code be implemented and how reliable for the system to be implemented.

Due to the fact mentioned above, a project development schedule is highly needed to ensure that effort is distributed within the prescribed time frame to make the best use of resources. Figure 1.2 illustrates the project development schedule for Digital Library of Theses.

No	Phase	Duration	Start	Finish	2000							2001		
					Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
1	Requirement Analysis	75 days	10.6.2000	31.8.2000	▨	▨	▨							
2	Quick Design	120 days	1.9.2000	15.1.2001				▨	▨	▨	▨	▨		
3	Construction of Prototype	130 days	1.9.2000	15.1.2001				▨	▨	▨	▨	▨		
4	Prototype Evaluation	130 days	1.9.2000	15.1.2001				▨	▨	▨	▨	▨		
5	Refine Requirement	130 days	1.9.2000	15.1.2001				▨	▨	▨	▨	▨		
6	Engineer Product	130 days	1.9.2000	15.1.2001				▨	▨	▨	▨	▨		
7	Documentation	70 days	20.12.2000	5.3.2001							▨	▨	▨	▨
8	Presentation Preparation	7 days	23.3.2001	1.3.2001										▨

Task ▨

Figure 1.2 Project Schedule

1.7 RESEARCH CONDUCTED

Research in the following areas was conducted:

- **General Digital Library of Theses basic requirements especially for administration**

  To find out the basic requirements (modules) for a digital library of theses, the research is mainly based on system administration part, which includes getting information on the digital library modules, research on existing digital library of Theses on Internet and so on.

- **Internet**

  To understand the structure of the Internet and Intranet and how the TCP/IP protocol serves as a bridge to enable the communication between clients and server.

- **Three-tier client/server architecture including database and web server**

  Research is earned out on how to implement a three-tier client/server system. It includes getting information on the implementation of three-tier client/server architecture, how to implement database server and also the web server in the three-tier architecture environment.

1.7 REPORT ORGANIZATION

This report contains the information on research and development of the Digital Library of Theses (Administration Side). It start out with chapter one that gives an project overview, the objectives of building this project and scope of the project. It also includes the project development schedule and its development strategy. Chapter two will discuss on the research done for the digital library system project, the relationship between digital library and the internet, the evolution of the theses system, current theses system, advantages of World Wide Web approach contribute to theses system and the other related issues of theses system. Chapter three describes the system analysis and the functional and non-functional requirement specifications of the system. Besides, it also includes the consideration of programming language and others. Chapter four describes the functional, user interface, and screen design issues.

Chapter Two

Literature Review

CHAPTER 2 LITERATURE REVIEW

2.1 TRADITIONAL THESES SYTEM

2.1.1 Introduction

The advancement of civilization has been paralleled by advancements in the means of expression and sharing of knowledge. The key documents of graduate studies are theses, dissertations, project reports, technical reports, and other similar forms. For the sake of brevity we will refer to them as TDs. These documents have been largely the concern of graduate students, faculty, and the research community. They also have been handled by libraries, which in effect "publish" them or for literature references.

Traditional theses system based on University of Malaya, Faculty Science Computer and Information Technology has been followed for a few years. Through out the year, all of the thesis done by graduates' students were collected and stored in faculty's document room for references and storage purposes. Figure below shows the traditional theses system implemented by faculty.

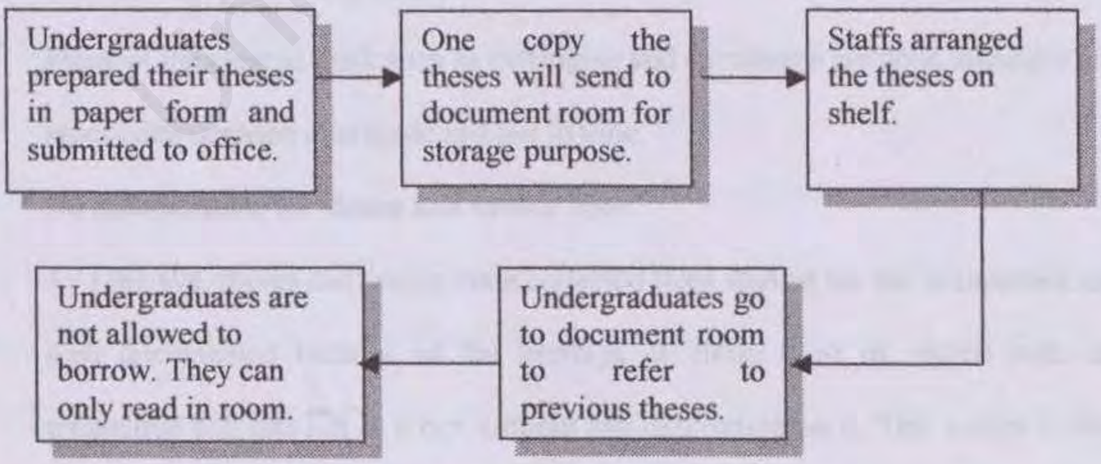

Undergraduates prepared their theses in paper form and submitted to office.
One copy the theses will send to document room for storage purpose.
Staffs arranged the theses on shelf.
Undergraduates go to document room to refer to previous theses.
Undergraduates are not allowed to borrow. They can only read in room.

Figure 2.1 Traditional theses system's diagram

2.1.2 Weakness and Drawbacks of Traditional Theses System

As we can see from the diagram, the undergraduates of faculty must go to document room in order to read or get the previous year's theses for referring. However, this has cause many inconveniences for both lecturers and student in faculty. As a result, unsatisfactory aroused among the students and lecturers. From the observation and general opinion by the users, below are several major problems regarding the current system:

- **Less timely public access to users**

  The document room only can be accessed in certain time slot during office hours.

- **Shelf space required for storage**

  Every year, hundreds of theses will be collected and these require more shelf space for storage as time going on.

- **Unsystematic arrangement of theses**

  The theses in document are not well-arrange because the student did not place back the theses accordingly and also due to the lack of clerk. As a result, student and lecturer in many times cannot find the theses they want.

- **Clerical and processing decision made manually**

  Many of the clerical work such as cataloging and circulation are done manually and processing decision also made just not in time.

- **No maintenance for theses and source code**

  As I believe, theses and source code collected from student are not maintained and well documented because of the shortage of clerk. Most of source code are unlabelled and just left in a box without any description on it. This makes it even more difficult to maintain.

- **Resources are not fully used**

  Resources are not fully used due to the consequences of no maintenance and some of the source codes are not reuse to benefit us.

- **Costs of preparing theses are high**

  Generally, the cost of preparing a thesis is around RM 25. Nevertheless, students have to pass up 4 copies of printed theses and therefore, costs of preparing theses are always high.

Due to the problem definition as listed above, the faculty rethink the way graduate students present their work and to make that work more important to the research community. The strategy is to require students to submit theses in digital form as the solution to the traditional theses system. Then create a database on server that provides free and instantaneous access to graduate students' research, which has traditionally been difficult to get.

2.2 ELECTRONIC THESES AND DISSERTATION

2.2.1 Introduction

Since the expansion of the Internet in particular, there have been particularly dramatic changes in the whole enterprise of research and education. The world of scholarship depends on people making their research available to others. When that is done electronically, more people can get access at lower cost, and more knowledge transfer occurs. This can stimulate education and research. It also can ensure that people give credit to the student for their work, and that the research is cited in others' publications. Before theses and dissertations were available electronically, not many were read. Electronic access multiplies the number of times works are read by a factor of ten or more. Some Electronic Theses Dissertations (ETDs) have been accessed thousands of times.

2.1.2 History of ETDs

In 1987, University of Microfilm International (UMI) hosted a workshop about electronic theses and dissertations (ETDs) in Ann Arbor, Michigan, USA, with a focus on applying SGML - an international standard that is the "parent" of HTML - to the description of research that might affect the handling of dissertations. Nick Altair, a vice president of UMI, was involved in the application of SGML to electronic documents. Following this workshop, Virginia Tech provided $5000 to develop an SGML Document Type Definition (DTD), in conjunction with SoftQuad, so this idea could be pursued. Unhappily, tools to help authors create SGML documents were expensive and harder to use than common word processors, so this approach was only pursued with low priority. In 1996, when the Southeastern Universities Research Association (SURA) provided about $90,000 to Virginia Tech to explore these matters further, project staff developed a Document Type Definition called ETD-ML, which has been repeatedly refined to be easy to use and yet powerful enough to capture the important metadata and structure of ETDs.

As technology developed to help make ETDs feasible, UMI made enlightened changes in its policies. It changed its agreements with students to only request non-exclusive license to archive and make copies on demand. It also defined a procedure by which electronic submissions could be made.

UMI worked with Virginia Tech and other interested parties to explore further about ETDs. Open meetings were held in 1992 and 1994 with at least ten universities sending representatives. Since these meetings began, the Council of Graduate Schools and the Coalition for Networked Information (CNI) have been strong supporters as well. For the last several years, there has been a morning technical discussion about ETDs hosted by CNI prior to their fall and spring meetings; discussions also take place at CGS meetings.

Since the beginning of 1997, UMI has been scanning in all new works received, creating a PDF file, which is essentially a wrapper around the black-and-white page images. These simple ETDs numbered over sixty-thousand as of spring 1998 and will grow at the rate of about five-thousand per month. While these can assist with access, they cannot convey what is expressed with gray scale or color, and miss details in works that use very small fonts or other symbols. Nevertheless, UMI's service is important and valuable, and marks a significant step forward toward a digital library of These Dissertations.

2.3 INITIATIVES

As electronic publishing technologies developed, a number of parties have thought about ETDs. Of particular importance are the students, who write the works and who often desire to use approaches that are more expressive than black ink on white paper.

2.3.1 Student Interest

The University of Virginia hosts a Web site about student efforts to create expressive ETDs. There are a number of students who have been interested in using new publishing approaches, hypertext, multimedia, hypermedia, and large volume storage units (e.g., CD-ROM), to prepare ETDs. Many of these are very creative, are more effective in expressing their results than usual, and so function as the real pioneers who help define the emerging ETD genre.

Since multimedia content types often require a great deal of storage space, one form of the push toward ETDs involves turning in CD-ROM. While technically one can use tape or network transfers to move such large volumes of data, CD-ROMs have several advantages:

- with good choice of media and care, they may last 10 years or more;
- many multimedia files (e.g., video) can be played off the CD-ROM, giving a similar effect to that of a book that can be read;
- it is easy to copy from a CD-ROM to other storage devices.

However, turning in a CD-ROM requires handling, and possibly storage of the media by a library, which may add to expenses relative to other approaches like network submission.

2.3.2 Other Related Efforts

Many other groups have been exploring ETDs. One such effort is to apply SGML. In particular, the many years of effort invested in the Text Encoding Initiative should bear fruit when students are working with texts marked up according to TEI guidelines. A pilot exploration of these issues was undertaken at the University of Michigan. Further work at Michigan, and also at University of Virginia, will investigate using the TEI DTD for creating ETDs as well as how well conversions can be make between that DTD and the ETD-ML form developed at Virginia Tech. Oxford University with long experience in electronic archives also involved in TEI.

2.3.3 VT ETD Initiative

While there are many ETD initiatives, as explained above, the most extensive effort has occurred at Virginia Tech. There is a campus Web page on this as well as additional information available from the Library. The Virginia Tech Graduate School, Library and Computing Center are committed for the long term to make ETDs the norm.

In addition to hardware and software to support ETDs, Virginia Tech also has a rich network infrastructure, including a Vbns (high speed Internet research and education backbone) connection through "Net.Work.Virginia," the statewide ATM network that it runs and which includes educational institutions all over the Commonwealth. Thanks to donations from IBM, one computer runs the IBM Digital Library product and will be used to provide access through that software, while another computer with 4 terabytes of storage is ready to handle even the largest multimedia files.

Virginia Tech's ETD-related information has been organized into four parts:

1. student submission (including policies, checklists, training materials, and automated scripts);
2. the searchable collection managed by the Library using OpenText software;
3. other copies of the searchable collection, also on Library equipment, used for demonstrations of digital library interoperability and to provide additional (experimental) services; and
4. information related to NDLTD (the project and its members).

2.4 STATISTICS

If we ignore page image collections like that of UMI, Virginia Tech's ETD collection is the largest. Thus, we explore some aspects of it, beginning with a discussion of statistics.

2.4.1 Statistics

Table 2.1 gives access statistics for the collection, based on data covering 1996 and other data covering 1997. It is clear that:

- there has been tremendous growth in access from one year to the next;
- there is 10-100 times as much access to this collection than occurs with regard to TDs on library shelves or even to
- works held by UMI; and
- accesses are from a large number of different sites.

1996 Value	1997 Value	Description
37,171	247,573	Total successful request
102	685	Av. successful request
4,600	72,854	PDF file downloads
28,225	129,831	HTML file downloads
9,015	22,725	Distinct hosts served
3,229	25,953	Gbytes transferred
9,038	73,574	Av. Mbytes transferred

Table 2.1 Access Statistics

Table 2.2 shows that accesses are from many domains. As expected, most accesses are from other educational institutions. However, a substantial number are from "commercial" organizations, possibly from industrial research labs. While it is possible that such accesses are really by people taking classes part-time, who use a computer in their place of work, it is likely that many are involved in corporate R&D.

Domain	1996	1997	% Increase
.edu	15,314	112,876	637 %
.com	5,309	48,540	814 %
.net	2,522	14,026	456 %
.org	375	3,132	735 %
.gov	282	1,362	383 %
.mil	188	1,872	896 %

Table 2.2 accesses by Internet Domain

Table 2.3 shows that accesses are from many parts of the world. The rows are ordered based on number of accesses, given in descending order. It also should be noted that there were vast increases in numbers from one year to the next. While there are many accesses from countries where English is spoken, many accesses are from countries where English is not as commonplace.

Country	1997	1996	Increase	% Increase
United Kingdom	2922	850	2072	244 %
Austria	2501	608	1893	311 %
Germany	2378	346	2032	587%
Canada	2367	713	1654	232 %

South Korea	1264	387	877	227 %
France	1161	463	698	151 %
Brazil	1130	183	947	517 %
Thailand	967	22	945	4295 %
Greece	958	83	875	1054 %
Netherlands	876	191	685	359 %
Portugal	768	68	700	1029 %
Italy	701	250	451	180 %
Singapore	662	230	432	188 %
Argentina	577	14	563	4021 %
Japan	495	101	394	390 %
Malaysia	436	59	377	639 %
Finland	424	255	169	66 %
Denmark	411	35	376	1074 %
Chile	409	18	391	172 %
Taiwan	311	40	271	678 %

Table 2.3 Download by country for 1996 and 1997 showing increases

Resources: Virginia Tech

2.5 CASE STUDIES

2.5.1 University Microfilm International (UMI)

When we talk about ETD, we cannot ignore the pioneer in this new evolution, UMI. UMI became the initiator of the movement toward digital dissertations by convening a meeting in1987 to discuss the concept. It has a representative on the NDLTD's steering and technology committees and has cooperated with Tech since the start of the project, continuing to be active with the NDLTD while developing its own service. In addition, UMI is committed to using and promoting the applications and standards developed by the NDLTD.

Beginning with 1997 dissertations, UMI's ProQuest Digital Dissertations program will assure digital formatting for all submissions, either by accepting dissertations in electronic format or by scanning and digitizing paper or microfilm submissions. Submissions in electronic format will be printed out and microfilmed, and the digital format will be entered into the digital archive for distribution online. Documents on CD-ROM will be distributed only on CDs. Compound documents, consisting of both text on paper and an electronic format, will be processed in a manner similar to paper documents but will be distributed as hardbound copies only. A library binding is considered the only secure way to accommodate printed text with a CD or floppy disk attached to the binding.

Institutions that do not subscribe to the UMI service pay $19.5 0 for a copy of the digital format of a dissertation (with a discount for additional titles). Those libraries that do subscribe receive the following:

- Web access to Dissertation Abstracts
- Preview of the first twenty-four pages of all digital dissertations
- Full-text copies. Access to all ETDs from the U.S. and Canada from 1997 onward is included in the subscription price. For downloading a dissertation, UMI opens up an FTP line for the requested title for forty-eight hours (with two, forty-eig it-hour extensions if necessary)
- Online access to MARC bibliographic records for the parent library.
- Consistent indexing and quality standards

- One easy-ordering interface

UMI may well be the choice of smaller institutions that can not easily implement their own publishing mechanisms. Indeed, even large institutions that see their current goal strictly as providing access to the scholarly record, might find UMI's vendor model preferable. Since UMI has only recently begun its digital service, its potential success in the marketplace is currently an imponderable.

2.5.2 West Virginia University (WVU)

From many sites on electronic theses system, WVU is one of the famous university using the electronic theses system and it is chosen as case study for further understanding on e-theses system. At WVU, all dissertations written in partial fulfillment of the requirements for any doctorate degree conferred by the University, and all theses written in partial fulfillment of the requirements of any masters degree conferred by the University must ordinarily be filed electronically with the WVU Library system according to its procedures for such filing. Exceptions to filing electronically must be approved by the Office of the Provost. Copyright to electronic theses and dissertations are subject to the appropriate provisions of the WVU Copyright Policy.

Once a student has successfully defended the thesis or dissertation, they submit the ETD online. The ETD is submitted in the form of one or more computer files in Adobe Acrobat format. This file form retains all formatting information about a document, while providing an electronic structure, which allows multi-media information to be viewed on any computer. The Acrobat format is generated by the program Adobe Distiller from a postscript file printed from the student's word processor. Once the ETD files are ready, they are submitted electronically through a web page that the student fills in with information on the thesis.

ETDs at WVU are made accessible through the World Wide Web. All ETDs have an abstract available for worldwide access. Three levels of access to the body of ETDs are possible at WVU. The bodies of most ETDs are openly available for reading worldwide. Some theses are made readable only by computers at WVU, an option available to protect some copyright restrictions imposed by publishers. A few ETDs are restricted from viewing by anyone for a one-year period in order to protect intellectual property rights.

2.5.3 NDLTD

Building on work that began in 1987, the Network Digital Library of Theses and Dissertations (NDLTD) was launched in 1996, and already has led to national project in USA, Australia and Germany, as well as involvement at universities in Canada, England, Hong Kong, India, Mexico and so on. It illustrates the move toward worldwide collaboration through digital library.

From the point of establishment of NDLTD, Virginia Tech who wish to extend this to the national level, proposed a broader effort, to improve graduate education through a National Digital Library of Theses and Dissertations, and received funding from the U.S. Department of Education (FIPSE). Then, as international members joined too, the name was changed to the Networked Digital Library of Theses and Dissertations (NDLTD). Additional support has been provided by may groups, especially Adobe, CNI, CGS, IBM, Microsoft, and OCLC

NDLTD calls for a sustainable, worldwide, collaborative, educational initiative of universities committed to encouraging students to prepare electronic documents and to use digital libraries. One of the key issues for NDLTD is to ensure that ETDs are preserved. At Virginia Tech the Library has committed to handle this, by copying to new media as that becomes available, by keeping multiple backups, and by migrating file formats as needed. For copyright issue, any conflicts that might arise with publisher policies can be handled by restricting access to the originating campus by using digital library access control.

2.5.4 Internet Library Administration System (NetLibrary)

The Internet Library Administration system is a system done by Phoon Koak Wai, graduate of UM, session 1997/1998. The administration system is made accessible through the World Wide Web. This web-based library administration system is developed for general library that enable librarians to maintain and also patrons to access the library's resources. The entire system is based on three-tiered client/server architecture. This system enables librarians to access to it from anywhere, anytime for remote cataloging, system maintenance and also circulation tasks.

The major modules of NetLibrary administration system are cataloging, which use to maintain the bibliographic records of the library; circulation, use to handle circulation transaction such as returning, renewing and reserving of items; user maintenance, calendar maintenance, security feature that allow authorized user only for accessing the system, dual languages for bibliographic records and on-line help.

2.5.5 Internet-Based Journal Management System- Electronic Journal General University (EJGU)

The EJGU system is a system done by Ooi Lee Chuen, graduate of UM, session 1998/1999. The system is a web-based system for managing the online journal. EJGU system enables subscribers and non-subscribers to access and read materials under different authentication through various Internet browsers. It also allows administrator to add in new types of journal. It consists of several commonly used functions such as article publication, subscriptions, online search engine editorial board and articles contributions. All articles have an abstract available for worldwide access, but full text can only accessed by authorized subscribers. The application in EJGU demonstrates the n-tier client/server architecture. Each client needs to establish a connection to the remote host by using the TCP/IP protocol.

2.5.6 Features Highlighting

The table shows the administration features comparison for the generally what we call e-theses system (NDLTD), e-journal system (EJGU) and e-library system (NetLibrary):

Features	NDLTD	EJGU	NetLibrary
Online help	X		X
Search engine	X	X	X
Editorial/ committee	X	X	
Cataloging	X		X
Publication/ Submission			
• Uploaded by administrator		X	X
• Uploaded by author	X		
• Posted by			

email/diskette/CD-ROM		X	
Announcement			
Archives	X		
Statistics	X		X
Control on access status	X		-
Copyright consideration	X		-
Submission Format	PDF	PDF	-
Platform	Unix	Windows NT	Active X platform

Table 2.4 Administration features comparison for e-theses system, e-journal system and e-library system

From the table, NDLTD and NetLibrary have online-help for guiding the user to use the function of system. Two of the systems, NDLTD and EJGU have committee or editorial board for controlling the quality of their publication while NetLibrary do not have any publication. However, NDLTD did cataloged the theses and dissertation like library system while collection of journal articles are not much as library and therefore it does not have cataloging function. For the way of submission, NDLTD enables authors to upload the thesis by themselves. EJGU and NetLibrary upload the thing through system administrator. EJGU also allows the people to send in their articles through email and diskette. Since long-term accessibility of electronic document is a critical issue, NDLTD has an archival function for the preservation of digital document. NDLTD also controls on access status of thesis by allowing student to choose one of the three-levels of access that are let anyone on the Web read a work, let only people with accounts at Virginia Tech read it or keep it off line entirely. PDF is most famous format for electronic document and is been used as format for submit in both NDLTD and EJGU.

2.6 BENEFITS OF ETD

From the literature review that been carried out, it is rational to have the ETD, which not only stimulate education and research but also bring other benefits as well as listed below:

- Access time and area more wide than before, more than 12 hours a day, six days a week, text delivery is immediate.
- The dissertation is ready for the students as soon as it is mounted on the server.
- Files are easily searchable and indexed.
- Authors can exercise more creativity, such as the use of interactive elements, hypertext, raw data file, virtual reality and multimedia.
- Students who create their own digital dissertations learn the basic skills of scholarly publishing in electronic format.
- Storage space is saved in the library, no staff are needed for circulation and reshelving.
- Library processing can be simplified through use of metadata mapped to *MARC* (author and title information, etc. is coded to correspond to fields In the bibliographic records), and records enhanced by the addition of the abstracts.
- Cooperative ventures in publishing and archiving can save universities costs in the future.

2.7 OTHER ISSUES

2.7.1 Archiving

Although switching from paper to electronic documents for theses and dissertations can save copying charges for students, shelf space in libraries, and shipping and handling expenses in universities, not everyone who has heard of the idea has been immediately amenable to this innovation. Although the greatest barriers to change are often fear, inertia and tradition, some universities considering the growing ETD initiative have hesitated primarily because of the real problem of the long-term accessibility of electronic documents.

Archiving and preservation of original digital documents is a thorny issue. The Association of Research Libraries (ARL) standards for archiving digital materials at this time specify one of two formats HTML or PDF No one has come up with satisfactory provisions for handling multimedia.

At UMI, they still consider microfilming to be its dependable archival medium. Tech runs a script (similar to a macro) that automatically generates an e-mail message to UMI whenever a new dissertation has been mounted, giving the URL author- and title. UMI then downloads whenever it chooses, prints, microfilms, and adds the file to its own digital database. Tech considers UMI to be its "emergency" backup, since it does not keep an archival paper copy itself. Michigan also intends to continue to deposit a copy of its dissertations with UMI, but more in the spirit of continuing to make its publications available to those institutions who may not be able to acquire digital material themselves.

In deciding on archival procedures, the following factors need to be considered

- Making and maintaining tape back-ups of ETDs should be standard operating procedure. however, magnetic media are not secure from data loss.
- Keeping an archival paper copy., at least for an initial period, could ease the transition to electronic format for those stakeholders who would resist abandoning print completely.

- Writing the dissertations to CD-ROM might provide more-secure long-term preservation.

For long-term preservation, which brings up the question of technological obsolescence, one must think of not only the content but also the medium. The two common methods of digital preservation are refreshing and migration, Refreshing data (copying the content periodically, such as from an old tape to a new one) is not likely to solve the long term problem. For complex files, critical functionality in the original file can eventually be lost. The technique of migration, however, holds more promise. It is possible to migrate older files through newer stages of technology when the new system has been designed to emulate the old so that it can accept the old files. Migration preserves the data and form of the original. Most current hardware and software designers take care to ensure those new versions will accept older files.

UMI assures that it will maintain the usability of digital files. The bibliographic utilities OCLC and RLG have begun to position themselves to provide digital storage services for members if a market develops. But the question remains: Who will pay for such a service? Will it be the members of the organization or the users? No one knows yet what new preservation techniques may develop or what the cost will be to ensure that ETDs remain intact and accessible.

2.7.2 Standard Electronic Formats

Agreement on the standard representations for ETDs is particularly important, for fewer authorized formats mean more-widespread compatibility and access, as well as fewer difficulties in developing effective archiving. The library can preserve the ETDs by copying to new media as that becomes available, by keeping multiple backups, and by migrating file formats as needed. This latter chore can be reduced somewhat if open standards are used. Thus, Table 1 indicates what formats are encouraged for files submitted as part of an ETD.

Type	Formats
Text	ASCII, SGML, PDF
Image	CGM, GIF, JPEG, PDF, PostScript, TIFF
Audio	AIF, MIDI, MPEG-2, WAV
Video	MPEG, QuickTime, Movie
Multimedia	Author Ware, AutoCAD, Director, Excel

Table 2.5 File Types and Formats

At this time the most popular format for ETDs is portable document format (PDF), although a number of publishers believe that SGML would be more flexible in time.

- **Adobe Acrobat.** Acrobat (PDF files) is software that allows documents created on a word processor to be made available on the Web. PDF files retain all the formatting and graphics of the original WordPerfect or Microsoft Word product. Larger and more-complex documents, especially in physics or mathematics, are often produced in LaTeX or TeX format, which use PostScript fonts because they are useful for rendering mathematical formulae. These can be converted to PDF files by using Adobe Distiller. Documents can be downloaded and read using the Acrobat Reader, obtainable at no charge on the Web from Adobe. PDF is a widely used format in electronic publishing.

- **SGML.** Standard Generalized Markup Language, which uses tag's to embed formatting codes in documents'. has great potential because it is platform independent, converts easily to other modes of presentation., and would handle multimedia better than PDF. The files are automatically converted to HTN/IL when accessed via the Web. One of the distinctive, and most useful, characteristics of this language is its application of the concept of *document type definition* (DTD), which defines particular structural units for a particular type of document. For example, a novel has an author, title, text as part of its DTD if it doesn't, it's not a novel to the computer. Likewise, a dissertation has an author, title, statement of degree-granting institution, abstract, etc.-If it doesn't, it's not a dissertation. This definition of a set of tags for a particular document type specifies exactly what the document will look like. The process is of course much more sophisticated than this simple example.

  Therein lies one difficulty in using SGML: there is not vet a completely reliable DTD for dissertations. Tech has developed one that it says is basically functional. but no template has been devised yet that would make it easily applicable by students. Probably the greatest disadvantage to SGML is its complexity. Even with a suitable DTD, it would remain difficult for most students to apply themselves, as there is no simple editing/authoring tool for turning a word processor document into SGML. A very recent development, XML (Extensible Markup Language), which is a simplified form of SGML, is becoming popular and may help to provide a solution to the difficulty.

2.8 CONCLUSION

Electronic Theses and Dissertations can help us share knowledge and culture on a large scale. That sharing can be even richer if linguistic variation is preserved as well, but that presents unique challenge for digital libraries to surmount. These problems are important to solve, so digital libraries can help enrich world culture and understanding.

It may be a key driving force for sharing knowledge and culture. If all theses are captured electronically, and most are freely shared, there will be tens of thousands of new works each year, many of which will deal with topics like history, sociology, linguistics, religion, and architecture. These will directly help people learn about other cultures, while more technical works will help readers learn about methods and approaches adopted by groups in remote locations. It is clear that it is here to stay, and will become more commonly used in upcoming years. This will also encourage the improvement of graduate education and the increase in knowledge sharing that can accompany use of ETDs.

Chapter Three

System Analysis

CHAPTER 3 SYSTEM ANALYSIS

3.1 INTRODUCTION

System analysis is an important activity when new information systems are being built or the existing ones are changed. System analysis is conducted for several purpose as listed below:

- Identify the user's requirements

  This is the most crucial role in this system analysis. The acceptability of the system after it has been delivered depends on how well it meets the user's need and supports the work to be automated.

- Evaluate the system for feasibility

  The analysis and study will decide if the proposed system will be cost-effective from business point of view and if it can be developed given existing budgetary constraints.

- Allocate functions to hardware, software, people, database and other system elements

3.2 REQUIREMENTS DEFINING'S TECHNIQUES

There are a few techniques used to define requirements. One of them is carrying out the literature review. A clearer perception of a scholarly electronic theses digital library is known well after the literature review. On top of that, Internet is the main resource in searching materials. Internet surfing techniques is a very useful and widely used for developer to seek several of information. Information on theses and dissertation, electronic theses system definition and specification, and also sample electronic theses system can be obtained from various website. However, there are other methods such as research, questionnaire and interview. Research on electronic theses system requirements was done by reviewing and referring related books to obtain relevant and useful information. Interview with a few lecturers in related field also carried out to further assist in understanding the concepts of the system and requirements. Mr. Phang, Mr. Ling Teck Chaw, Mr. Teh Kang Hai, Dr. Diljit and Puan Zainab are interviewees in the interview session and have provided valuable and precious advice on the topics.

3.3 SYSTEM MODEL

System model shows the relationships between the system components and the system and its environment. Digital Library of Thesis allows administrator to add in new faculty, new department for theses publication. It consists several commonly used functions such as theses publication, maintenance and user access and queries. The application in this system demonstrates the n-tier client/server architecture. Due to it is a web-based application system, each client needs to establish a connection to remote host by using TCP/IP protocol. The system only contains 3 main modules as show below:

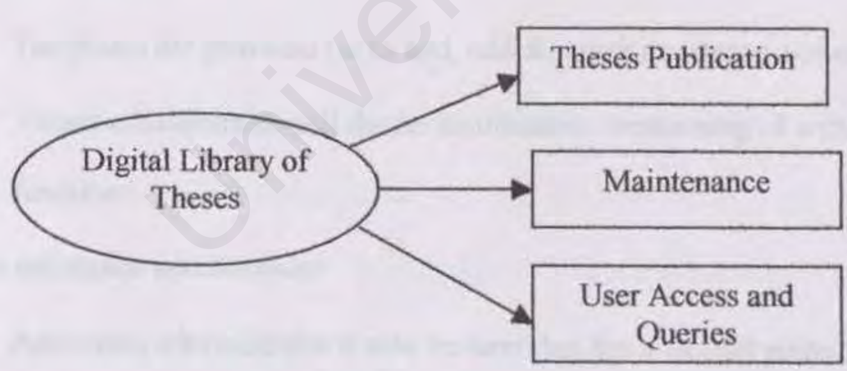


Figure 3.1 Digital Library of Theses system model

3.4 FUNCTIONAL REQUIREMENTS

System requirements set out what the system should do. A requirement may be functional requirement, that is, it describes a system service or function and alternatively, it may be a non-functional requirement, which is a constraint placed on the system or on the development process. The following are the functional requirements for Digital Library of Theses.

3.4.1 Thesis publication

For super administrator

- Templates are provided for super administrator to create and delete faculty.
- Templates also provided to add, edit & delete theses administrator in order to control the security of the system.

For theses administrator

- Templates are provided for theses administrator to create and delete department in faculty itself.
- Templates are provided for to add, edit & delete assistance administrator.
- Theses administrator will do the maintenance concerning of archives/backup function.

For assistance administrator

- Assistance administrator is also lecturer that has a student under their supervision. Templates are provided to add, edit and delete author and password.
- Assistance administrator must create one password to all students or one password for each student under them for uploading purposes.

- Template is provided for administrator to see the list of the student under them after registration of student.

3.4.2 Maintenance

- Templates are provided to create and delete announcement done by theses administrator and assistance administrator.
- Backup and archive will be done automatically by the system.

3.4.3 User Access and Queries

- Templates are created for Super administrator, theses administrator and assistance administrators to change or update their personal information including password, login name and email.
- Report on student, assistance administrator and theses administrator or others will be generated for controlling and displaying useful information to concerning administrator.

3.5 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe a restriction on a system that limits the choices for constructing a solution to the problems and the set of standards, which a delivered system should meet. The non-functional requirements in Digital Library of Theses are as below:

3.5.1 Graphical User Interface (GUI)

The system has provided a frame (toolbar) wherever user is and eases the user to access to any part of the system. The system also provides a user friendly interface with WIMP

(Window, Icon, Menu, and Pointing Device) features. A simple language icon is used for a better understanding of the function. Any request can be posted to the administrator for further information through e-mail.

3.5.2 Reliability

A reliability system should be consistence while functioning. It should be run smoothly although there are many web users using the system simultaneously. The system should not produce dangerous or costly failures when it is used in a reasonable manner.

3.5.3 Robustness

This system should be able to check the input validation before it continues for further data processing to avoid unnecessary disaster. When there is any mistake detected, this system will prompt an error message to indicate the mistake. For example, when a user does not key in value for a mandatory field, an error message ill be prompted.

3.5.4 Maintainability

This system should be easily to be maintained, simplicity and effective. To reduce power manpower involvement, templates are provided for administrators to manage this system.

3.5.5 Correctness

Correctness includes traceability, completeness and consistency. The logic of the system must be able to be traced. In this system, comments are written in the scripts. In addition, performance of this system should be consistence and stable. This system will also check the user authorization and authentication.

3.6 PLATFORM AND WEB SERVER CONSIDERATION

3.6.1 Microsoft Windows NT Server

Microsoft Windows NT Server is a complete and powerful platform that provides server operating system. It provides the backbone for a complete, organic system, where all elements working together seamlessly. When it is joined with other Windows NT – related products, including the BackOffice® family of applications and Windows NT workstation, Windows NT Server provides the foundation for a powerful and well integrated environment. The introduction of new management tools in Windows NT Server and the Option Pack has provided great assistance in setting up websites, managing the content and analyzing usage patterns for improvement. It serves as platform to publish and share information in a secure way over Intranet and Internet.

3.6.2 Microsoft Internet Information Server 3.0 (IIS)

Microsoft Internet Information Server 3.0 is the only World Wide Web server that is tightly integrated with the Microsoft Windows NT Server operating system and is designed to deliver a wide range of Internet and Intranet server capabilities. IIS 3.0 was designed to deliver on the following objectives:

- **Integration with Windows NT Server**

  Because of the tight integration with Windows NT Server, IIS is easy to setup and manage, fast and secure.

- **Comprehensive Web Server Solution**

  IIS includes a built-in search engine, streaming multimedia capabilities, rich log file analysis tools, and much more.

- **Easy to develop, powerful Web-based applications**

  IIS introduces Active Server Pages, which make posting dynamic content and development of web based application easy.

3.6.3 UNIX Operating System

3.6.3.1 Introduction

The UNIX operating system was designed to let a number of programmers access the computer at the same time and share its resources. The operating system controls all of the commands from all of the keyboards and all of the data being generated, and permits each user to believe he or she is the only person working on the computer. This real-time sharing of resources make UNIX one of the most powerful operating systems ever. Although UNIX was developed by programmers for programmers, it provides an environment so powerful and flexible that it is found in businesses, sciences, academia, and industry. Many telecommunications switches and transmission systems also are controlled by administration and maintenance systems based on UNIX.

3.6.3.2 The uniqueness of UNIX

The features that made UNIX a hit from the start are:

- **Multitasking capability**

UNIX lets a computer do several things at once, such as printing out one file while the user edits another file.

- **Multiuser capability**

The same design that permits multitasking permits multiple users to use the computer. The computer can take the commands of a number of users -- determined by the design of the computer -- to run programs, access files, and print documents at the same time.

- **Portability**

A major contribution of the UNIX system was its portability, permitting it to move from one brand of computer to another with a minimum of code changes.

- **UNIX programs**

UNIX comes with hundreds of programs that can divided into two classes: Integral utilities that are absolutely necessary for the operation of the computer, such as the command interpreter, and tools that aren't necessary for the operation of UNIX but provide the user with additional capabilities, such as typesetting capabilities and e-mail.

3.6.3.3 Sun WebServer Solaris Operating System

Sun WebServer delivers security, reliability, manageability, and flexibility to drive down costs and deliver profitable new services to business and consumer markets. It optimizes the power of the Solaris Operating Environment and Solaris ISP (Internet Service Provider) Server, and is designed for the most demanding Internet environments. The Solaris Operating Environment optimized for Java computing. It is superior reliability, availability extensive scalability and high performance. Solaris ISP Server software is:

- Highly manageable and scalable, enabling ISPs to focus on delivering new services to increase revenue and margins Secure, delivering enhanced security with OS hardening and other features to provide superior protection against hackers and intruders
- Integrated, providing services that are tested to work together so that ISPs can use their scarce resources to develop and deliver new value-added services
- The features of Sun WebServer 2.1 are:
- Full HTTP 1.1 compliance.

- Java servlet (Java Servlet Development Kit 2.0) support.
- Scalability and performance are increased while administration costs are reduced. Improved graphical Web-based and/or command-line driven administration for easy access and configuration.
- Security through SSL, plus access control to prevent unauthorized use.
- Scalable virtual-hosting, supporting multiple Web sites on a single server. Domain-based virtual hosting allows the number of IP addresses does not limit the highest capacity and flexibility host names.

3.6.4 Preference of Operating System and Web Server

Windows NT Server is more preferable than UNIX for the following reasons:

- Overall, Windows NT Server is easier to manage than UNIX system
- There is a lot of supporting software for Windows NT especially the free downloadable option packs.
- With the Windows-based GUI, it is easier to accept and learn by users.
- Part of the Windows NT Server installation, which means that IIS is up and running with Windows NT Server.
- Windows NT Server 4.0 is 4 times faster as a web server than Solaris 2.6 with Sun Web Server 1.0 and has 10.3 times better price and performance.

3.7 PROGRAMMING LANGUAGES AND COMPONENTS CONSIDERATION

Digital Library Theses System is a web application that requires dynamic data retrieving to create informative system for its users. As a result, the consideration of programming languages and components used in implementation of this system are mainly based on the programming languages that can cater the needs to build dynamic web applications.

In order to build dynamic web-applications the concept of component framework standard is introduced to feed the web applications with interoperability, reusability and cross-platform. Many programming and components tools such as Microsoft Visual Basic, ActiveX, Java, Active Server Pages (ASP), Common Gateway Interface (CGI) do support the development of component framework that mostly based on two popular components standard, namely Component Object Model (COM)/ Distributed COM, and Common Object Request Broker Architecture (CORBA).

3.7.1 Component Object Model (COM)

Microsoft's multitier strategy is founded on a core technology known as the *Component Object Model (COM)*, which is a general architecture for component software. This means that it is a standard, not an implementation. The term *COM* means many different things to many people. On the one hand, it's a specification for writing reusable software that runs in component-based systems. On the other hand, it's a sophisticated infrastructure that allows clients and objects to communicate across process and computer boundaries.

The COM programming model is based on the distribution of class code in binary components. This means that software (components) that adheres to COM can be reused without any dependencies on source code. Developers can ship their work as binary files without revealing their proprietary algorithms. The reuse of code in a binary form also eliminates many compile-time problems that occur when applications are assembled using a development style based on source code reuse. COM is founded on the idea of *interface-based programming,* which means that COM have a formalized separation of interface from implementation.

3.7.1.1 Objects and Interfaces

What is an object? An object is an instantiation of some *class*. At a generic level, a "class" is the definition of a set of related data and capabilities grouped together for some distinguishable common purpose. The purpose is generally to provide some service to "things" outside the object, namely clients that want to make use of those services. An object that conforms to COM is a special manifestation of this definition of object. A COM object appears in memory much like a C++ object. Unlike C++ objects, however, a client never has direct access to the COM object in its entirety. Instead, clients always access the object through clearly defined contracts: the interfaces that the object supports, *and only those interfaces.*

When an object "implements an interface" that object implements each member function of the interface and provides pointers to those functions to COM. COM then makes those functions available to any client who asks. This terminology is used in this document to refer to the object as the important element in the discussion. An equivalent term is an "interface on an object" which means the object implements the interface but the main subject of discussion is the interface instead of the object.

3.7.1.2 Clients, Servers, and Object Implementors

The interaction between objects and the users of those objects in COM is based on a client/server model. There are also *object implementors*, or some program structure that implements an object of some kind with one or more interfaces on that object. Sometimes a client wishes to provide a mechanism for an object to call back to the client when specific events occur. In such cases, COM specifies that the client itself implements an object and hands that object's first interface pointer to the other object outside the client. In that sense, both sides are clients, both sides are servers in some way.

- **Object**

A unit of functionality that implements one or more interfaces to expose that functionality.

- **Object Implementor**

Any piece of code, such as an application, that has implemented an object with any interfaces for any reason. The object is simply a means to expose functions outside the particular application such that outside agents can call those functions. Use of "object" by itself implies an object found in some "object implementor" unless stated otherwise.

- **Client**

There are two definitions of this word. The general definition is any piece of code that is using the services of some object, wherever that object might be implemented. A client of this sort is also called an "object user." The second definition is the active agent (an application) that drives the flow of operation between itself an other objects and uses specific COM "implementation locator" services to instantiate or creates objects through servers of various object classes.

- **Server**

A piece of codes that structure an object class in a specific fashion and assigns that class a COM class identifier. This enables a client to pass the class identifier to COM and ask for an object of that class. COM is able to load and run the server code, ask the server to create an object of the class, and connect that new object to the client. A server is specifically the necessary structure around an object that serves the object to the rest of the system and associates the class identifier: a server is not the object itself.

3.7.1.3 Server Flavors: In-Process and Out-Of-Process

Any specific server can be implemented in one of a number of flavors depending on the structure of the code module and its relationship to the client process that will be using it. A server is either "in-process" which means it's code executes in the same process space as the client, or "out-of-process" which means it runs in another process on the same machine or in another process on a remote machine. These three types of servers are called "in-process," "local," and "remote" as defined below:

- **In-Process Server**

  A server that can be loaded into the client's process space and serves "in-process objects." Under Microsoft Windows and Microsoft Windows NT, these are implemented as "dynamic link libraries" or DLLs. This specification uses **DLL** as a generic term to describe any piece of code that can be loaded in this fashion which will, of course, differ between operating systems.

- **Local Server**

  A server that runs in a separate process on the same machine as the client and serves "local objects." This type of server is another complete application of its own thus defining the separate **process**. This specification uses the terms "**EXE**" or "**executable**" to describe an application that runs in its own process as opposed to a DLL which must be loaded into an existing process.

- **Remote Server**

  A server that runs on a separate machine and therefore always runs in another process as well to serve "**remote** objects." Remote servers may be implemented in either DLLs or EXEs; if a remote server is implemented in a DLL, a surrogate process will be created for it on the remote machine.

3.7.2 Distributed Common Object Model (DCOM)

3.7.2.1 Introduction

The Microsoft® Distributed Component Object Model (DCOM) extends the Component Object Model (COM) to support communication among objects on different computers—on a local area network (LAN), a wide area network (WAN), or even the Internet.

3.7.2.2 The DCOM Architecture

DCOM is an extension of the Component Object Model (COM). COM defines how components and their clients interact. This interaction is defined such that the client and the component can connect without the need of any intermediary system component. The client calls methods in the component without any overhead whatsoever. In today's operating systems, processes are shielded from each other. A client that needs to communicate with a component in another process cannot call the component directly, but has to use some form of interprocess communication provided by the operating system. COM provides this communication in a completely transparent fashion: it intercepts calls from the client and forwards them to the component in another process. Figure below illustrates how the COM/DCOM run-time libraries provide the link between client and component.

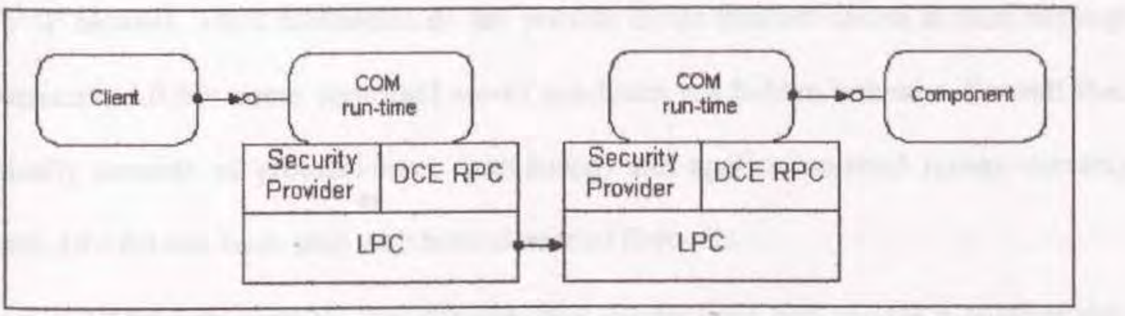


Figure 3.2 COM components in different processes

When client and component reside on different machines, DCOM simply replaces the local interprocess communication with a network protocol. Neither the client nor the component is aware that the wire that connects them has just become a little longer.

Figure 3 shows the overall DCOM architecture: The COM run-time provides object-oriented services to clients and components and uses RPC and the security provider to generate standard network packets that conform to the DCOM wire-protocol standard.

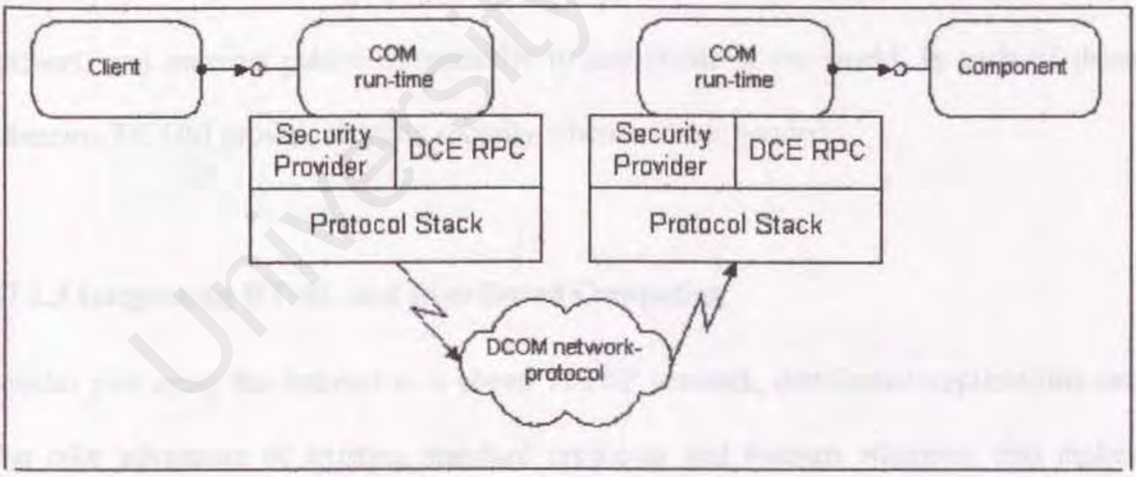


Figure 3.3 DCOM: COM components on different machines

3.7.2.3 DCOM Over the Internet

Since DCOM is an inherently secure protocol, it can be used without being encapsulated in a virtual private network: DCOM applications can simply use the cheap, global

TCP/IP network. Most companies do not provide direct Internet access to their desktop computers. All but some dedicated server machines are hidden behind a firewall that typically consists of protocol-level (port-based) and application-level (proxy servers) filters. DCOM can work well with both classes of firewalls:

- DCOM uses a single port for initiating connections and assigns a configurable range of ports to the actual components running on a machine.
- Application-level proxies can easily be built. They can be either generic (forwarding configurable DCOM activation and method calls) or application-specific.
- Server administrators can also choose to tunnel DCOM through HTTP, effectively bypassing most of today's firewalls.

With this range of options, DCOM-based applications can use the Internet for private connectivity within a company, private communication with external customers and partners, and massive public connectivity to any client in the world. In each of these scenarios, DCOM provide flexible security whenever it is needed.

3.7.2.3 Integrating HTML and Distributed Computing

Besides just using the Internet as a cheap TCP/IP network, distributed applications can also take advantage of existing standard protocols and formats wherever that makes sense. For non-interactive, textual, or simple graphical information, HTML pages are a great vehicle that provide a well known and easy way for users to access the information they require.

For more complex, structured, and interactive information, custom components can extend the HTML page to perform the real tasks of the distributed applications in a user-friendly, secure, and efficient manner. Simple business rules can be applied on the client side, providing immediate feedback to the user. More complex business rules can transparently involve server components over DCOM. Thanks to DCOM's language neutrality, these components can be implemented in virtually any programming language, including C++, Java, Visual Basic, or Cobol. Existing off-the-shelf components (ActiveX controls) can be tied to client-side or even server-side custom components using Visual Basic Script or JScript.

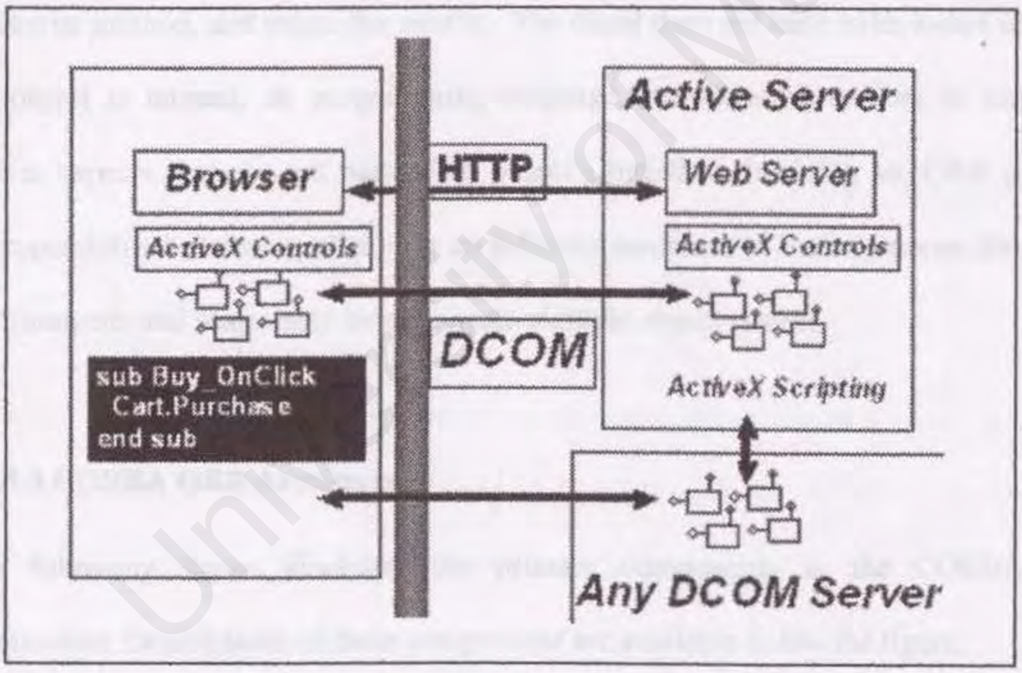


Figure 3.4 Integrating HTML and distributed computing

3.7.3 Common Object Request Broker Architecture (CORBA)

3.7.3.1 Introduction

The Common Object Request Broker Architecture (CORBA) is an emerging open distributed object-computing infrastructure being standardized by the Object

Management Group. CORBA automates many common network programming tasks such as object registration, location, and activation, request demultiplexing, framing and error-handling, parameter marshalling and demarshalling, and operation dispatching.

3.7.3.2 Object Request Broker (ORB)

The ORB is the middleware that establishes the client-server relationship between objects. Using ORB, a client can transparently invoke method on a server object, which can be on the same machine or across a network. The ORB intercepts the call and is responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results. The client does not have to be aware of where the object is located, its programming language, its operating system, or any other system aspects that are not part of an object's interface. In doing so, ORB provides interoperability between applications on different machines in heterogeneous distributed environments and seamlessly interconnects multiple object system.

3.7.3.3 CORBA ORB Architecture

The following figure illustrates the primary components in the CORBA ORB architecture. Descriptions of these components are available below the figure.

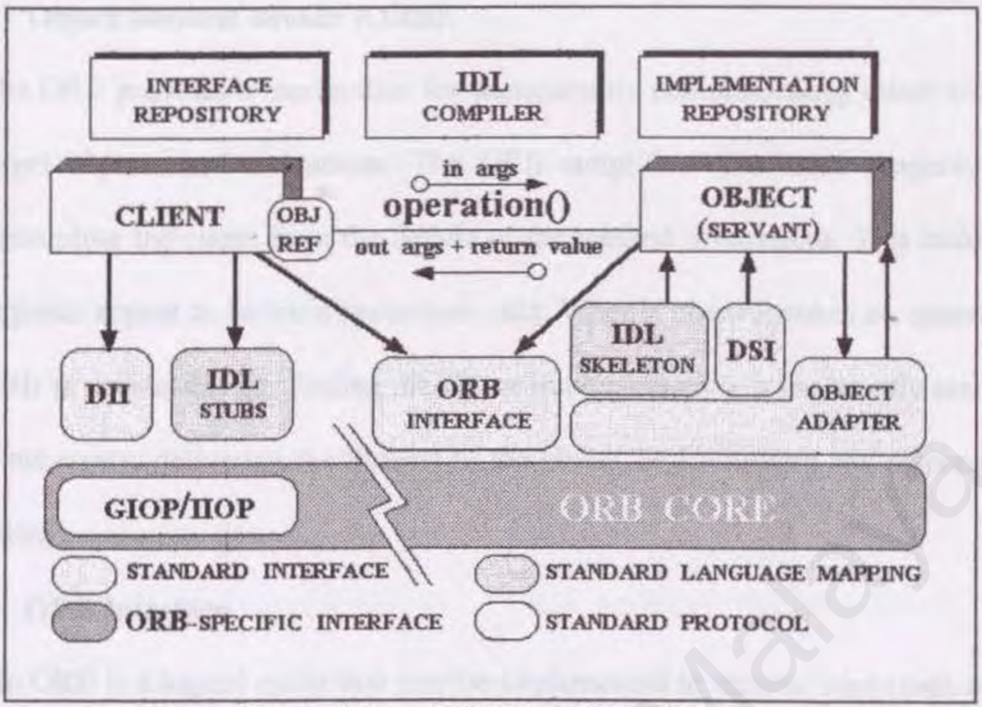


Figure 3.5 CORBA ORB Architecture

- **Object**

This is a CORBA programming entity that consists of an *identity*, an *interface*, and an *implementation*, which is known as a *Servant*.

- **Servant**

This is an implementation programming language entity that defines the operations that support a CORBA IDL interface. Servants can be written in a variety of languages, including C, C++, Java, Smalltalk, and Ada.

- **Client**

This is the program entity that invokes an operation on an object implementation. Accessing the services of a remote object should be transparent to the caller. Ideally, it should be as simple as calling a method on an object, i.e., `obj->op(args)`. The remaining components in Figure 3.6 help to support this level of transparency.

- **Object Request Broker (ORB)**

The ORB provides a mechanism for transparently communicating client requests to target object implementations. The ORB simplifies distributed programming by decoupling the client from the details of the method invocations. This makes client requests appear to be local procedure calls. When a client invokes an operation, the ORB is responsible for finding the object implementation, transparently activating it if necessary, delivering the request to the object, and returning any response to the caller.

- **ORB Interface**

An ORB is a logical entity that may be implemented in various ways (such as one or more processes or a set of libraries). To decouple applications from implementation details, the CORBA specification defines an abstract interface for an ORB. This interface provides various helper functions such as converting object references to strings and vice versa, and creating argument lists for requests made through the dynamic invocation interface described below.

- **CORBA IDL stubs and skeletons**

CORBA IDL stubs and skeletons serve as the ``glue" between the client and server applications, respectively, and the ORB. The transformation between CORBA IDL definitions and the target programming language is automated by a CORBA IDL compiler. The use of a compiler reduces the potential for inconsistencies between client stubs and server skeletons and increases opportunities for automated compiler optimizations.

- **Dynamic Invocation Interface (DII)**

This interface allows a client to directly access the underlying request mechanisms provided by an ORB. Applications use the DII to dynamically issue requests to objects without requiring IDL interface-specific stubs to be linked in. Unlike IDL stubs (which only allow RPC-style requests), the DII also allows clients to make non-blocking *deferred synchronous* (separate send and receive operations) and *oneway* (send-only) calls.

- **Dynamic Skeleton Interface (DSI)**

This is the server side's analogue to the client side's DII. The DSI allows an ORB to deliver requests to an object implementation that does not have compile-time knowledge of the type of the object it is implementing. The client making the request has no idea whether the implementation is using the type-specific IDL skeletons or is using the dynamic skeletons.

- **Object Adapter**

This assists the ORB with delivering requests to the object and with activating the object. More importantly, an object adapter associates object implementations with the ORB. Object adapters can be specialized to provide support for certain object implementation styles (such as OODB object adapters for persistence and library object adapters for non-remote objects).

3.7.4 COM/DCOM vs. CORBA

Both DCOM and CORBA provide good middleware solutions for three-tier distributed database systems. DCOM is better suited to handle the desktop and Microsoft Server needs. CORBA is the only solution for a heterogeneous system. There is a place where the desktop and the heterogeneous system will meet and there must be interoperability between CORBA and DCOM for this technology to succeed. This need can be addressed by the CORBA/COM bridge. The bridges enable CORBA systems to integrate COM desktops. Currently, the Object Bridge from Visual Edge Software is the only product that provides complete bi-directional communication between COM/DCOM and CORBA. Most other packages only allow CORBA to OLE communication. OLE automation requires dynamic invocation of CORBA methods that is quite expensive. Other makers of bridges are IONA Technologies, NobleNet, and Expersoft.

3.7.5 ActiveX

3.7.5.1 Introduction

ActiveX refer to a set of "strategic" object–oriented program technologies and tools from Microsoft for creating interactive web applications. These advances include client technologies, server technologies, tools and applications. It is a language-independent interoperability technology that enables software components written in different languages to work together in network environments. Three main components make up the ActiveX client technologies: ActiveX Controls, ActiveX Documents, and ActiveX Scripting.

- **ActiveX Controls**

ActiveX Controls function just like traditional OLE Controls (OCXs): They can reside locally on a client machine or they can be downloaded from the Internet. You can use ActiveX Controls to handle client-side interactions with the user. An ActiveX Control may be as simple as a button or as complex as a reporting tool. Since the same ActiveX Controls can also be used inside Visual Basic, you can expect to see a proliferation of

ActiveX Controls that you'll be able to use in your Web pages. An ActiveX control is roughly equivalent in concept and implementation to the Java Bean.

- **ActiveX Scripting**

Developers can use ActiveX Scripting to coordinate the activities of ActiveX Controls on their Web pages. Currently available ActiveX Scripting languages are VBScript and JavaScript. These scripting languages use OLE Automation to communicate with an ActiveX Control; the ActiveX Control fires OLE events that can be handled in the ActiveX Script code in response to user interactions (such as clicking on a button in the ActiveX control).

- **ActiveX Documents**

ActiveX Documents are an exciting piece of technology. Using Microsoft Internet Explorer 3.0, you can view active documents as well as HTML pages. In addition, you can allow users to navigate, view, and edit the contents of documents that can't be rendered using straight HTML alone--all within a Web browser, thus providing users with a consistent interface.

3.7.5.2 ActiveX Platform

Active platform uses ActiveX controls as a mechanism to interact with users and to automate everything from COM-enabled transaction to transaction processing (TP) monitors to Web servers. It has two parts: Active Server and Active Client.

- **Active Server**

It is really the middle tier that providing the location for business logic and primary application processing using components or Active Server pages (ASP). The core technology of Active Server including NT server, Microsoft Transaction Server (MTS), data management services, directory services, Web services, and network services. The

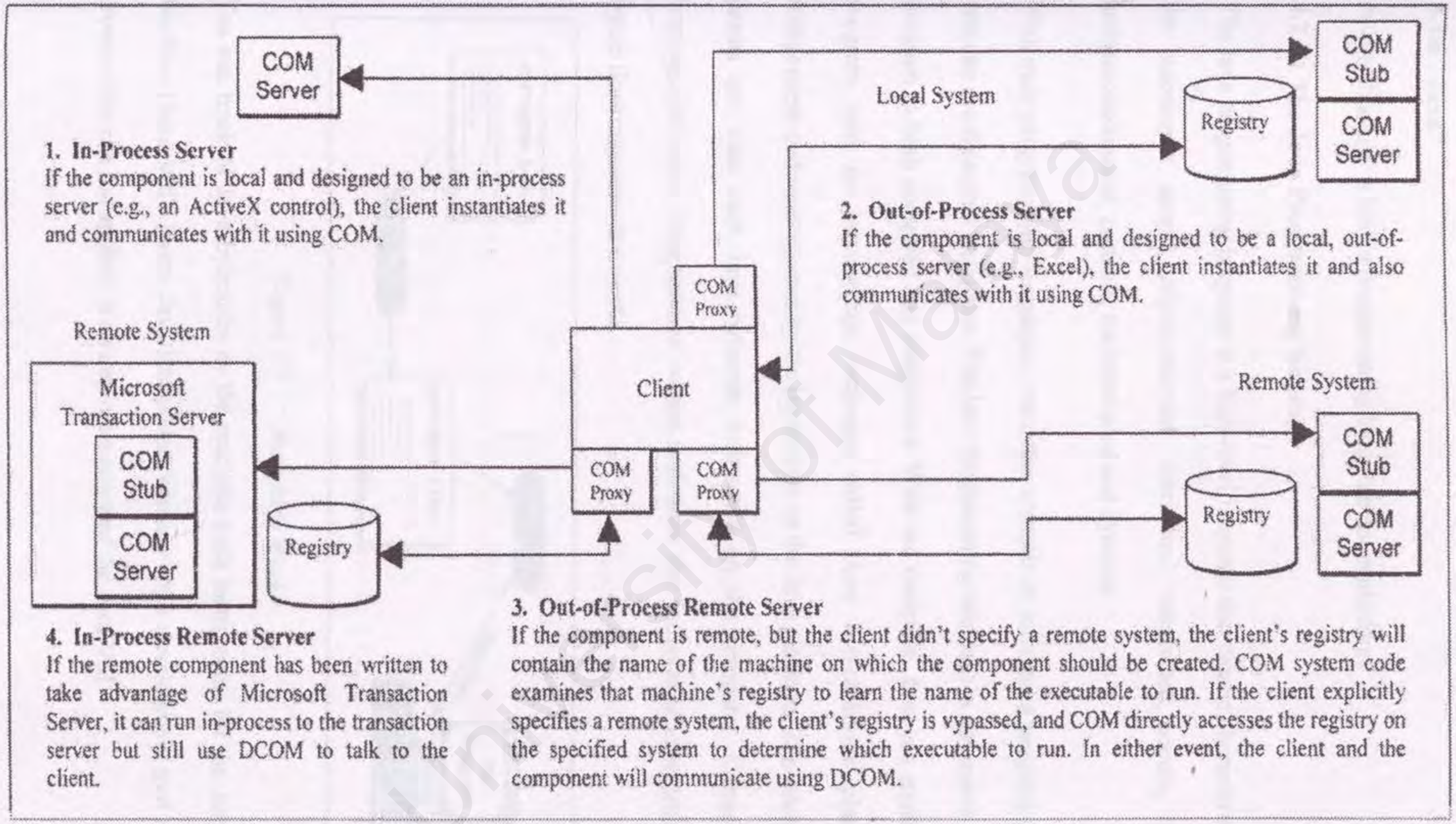
Web services are built around Internet Information Server (IIS), providing scripting mechanisms through ASP for server-side Web application development. Network services are built around DCOM, allowing the controls to connect over the network through the synchronous MS-RPC.

- **Active Client**

It is a cross-platform and is hoping to deliver technology to a broad range of operating systems, albeit on its own terms. The plan is to use standard HTML, Microsoft flavor of Java virtual machine (JVM), and a scripting engine using Microsoft Vbscript and Jscript (Microsoft's version of JavaScript).

3.7.5.3 The Four Way of ActiveX

The core technologies that enable ActiveX include COM/DCOM (the object model), OjectRPC (the transport), the registry (a database of component locations), monikers, automation, and structures storage. All ActiveX components can be started the same way.



1. In-Process Server
If the component is local and designed to be an in-process server (e.g., an ActiveX control), the client instantiates it and communicates with it using COM.

2. Out-of-Process Server
If the component is local and designed to be a local, out-of-process server (e.g., Excel), the client instantiates it and also communicates with it using COM.

3. Out-of-Process Remote Server
If the component is remote, but the client didn't specify a remote system, the client's registry will contain the name of the machine on which the component should be created. COM system code examines that machine's registry to learn the name of the executable to run. If the client explicitly specifies a remote system, the client's registry is vypassed, and COM directly accesses the registry on the specified system to determine which executable to run. In either event, the client and the component will communicate using DCOM.

4. In-Process Remote Server
If the remote component has been written to take advantage of Microsoft Transaction Server, it can run in-process to the transaction server but still use DCOM to talk to the client.

Figure 3.6 Four ways of ActiveX

3.7.6 Java

Java technology is both a programming language and a platform.

3.7.6.1 The Java Programming language

The Java programming language is a high-level language that can be characterized by the buzzwords: simple, object-oriented, distributed, interpreted, secure, robust, architecture-neutral, portable, multithreaded and dynamic.

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called *Java bytecodes*-the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java bytecode instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how this works.

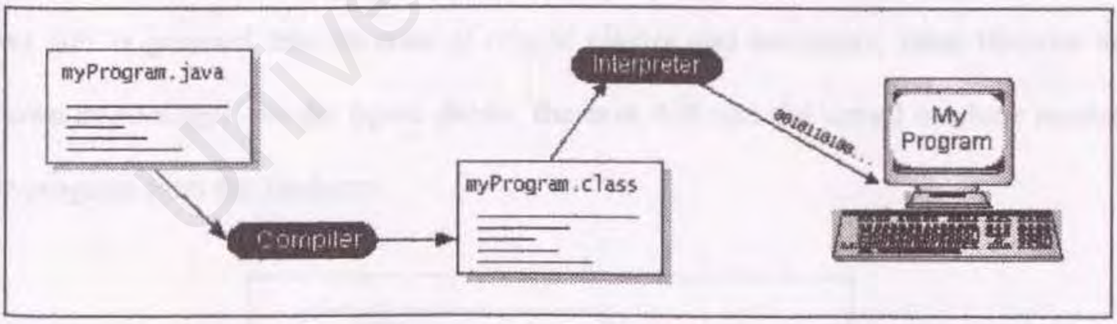


Figure 3.7 How Java Works

You can think of Java bytecodes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM.

Java bytecodes help make "write once, run anywhere" possible. You can compile your program into bytecodes on any platform that has a Java compiler. The bytecodes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

3.7.6.2 Java Platform

The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

Java VM is the base for the Java platform and is ported onto various hardware-based platforms. The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as *packages*. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.

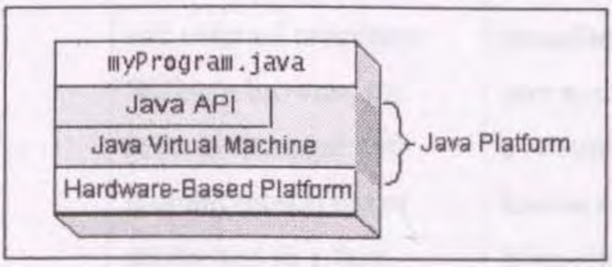


Figure 3.8 Java API and visual machine

3.7.6.3 Java Beans

Java Beans is a portable, platform-independent component model written in the Java programming language, developed in collaboration with industry leaders. It enables developers to write reusable components once and run them anywhere -- benefiting from the platform-independent power of Java technology. Java Beans acts as a Bridge between proprietary component models and provides a seamless and powerful means for developers to build components that run in ActiveX container applications. Java Beans components (or "Beans") can be used to give World Wide Web pages (or other application) interactive capabilities such as computing interest rates or varying page content based on user or browser characteristics.

3.7.7 ActiveX vs. Java

Below is the comparison between major features of ActiveX and Java:

Feature	Java	ActiveX
Safety - Protection from viruses or malicious code downloaded automatically across the Internet	The inbuilt Java security manager is consulted every time a Java application tries to access the disk, the network ports and external processes. Within a browser, the security manager will prohibit most types of access and so a Java applet downloaded off the Internet cannot cause any damage. Unfortunately,	ActiveX controls attempt to be safe by being "digitally signed" with an authentication code identifying the manufacturer. This allows a user to choose to download a control written by the well known and trusted Microsoft but not an equally legitimate control written by the lesser known Spiral Software. The ActiveX

	the security manager prevents an applet doing a lot of useful things too, so in time it is likely that the security manager will be weakened.	author is required to get a certificate to enable signing of controls.
Speed on a standard PC	Java is currently slow. This is a disaster for animations. Of course, Java should be a lot faster on Sun's JavaStation or any other system designed to run Java.	ActiveX has a excellent performance on a standard PC.
Ease of programming	Java is an elegant language but still somewhat specialised and requiring professional programmers.	ActiveX controls have traditionally been written in C/C++. However, Microsoft's Visual Basic 6.0 will allow ActiveX controls to be produced very easily, but only for the Windows95/NT operating system.
Portability - will the applet, control or application run on many types of machine?	Java will run unmodified on any computer that supports the Java virtual machine.	No. ActiveX controls have to be specially written for each machine and operating system.

Table 3.1 Comparison between major features for Java and ActiveX

3.7.8 Active Server Pages (ASP)

ASP is also an abbreviation for application service provider. An Active Server Page (ASP) is an HTML page that includes one or more scripts (small embedded programs) that are processed on a Microsoft Web server before the page is sent to the user. An ASP is somewhat similar to a sever-side include or a common gateway interface (CGI) application in that all involve programs that run on the server, usually tailoring a page for the user. Typically, the script in the Web page at the server uses input received as the result of the user's request for the page to access data from a database and then builds or customizes the page on the fly before sending it to the requestor.

ASP is a feature of the Microsoft Internet Information Server (IIS), but, since the server-side script is just building a regular HTML page, it can be delivered to almost any browser. You can create an ASP file by including a script written in VBScript or Jscript in an HTML file or by using ActiveX Data Objects (ADO) program statements in the HTML file. You name the HTML file with the ".asp" file suffix. Microsoft recommends the use of the server-side ASP rather than a client-side script, where there is actually a choice, because the server-side script will result in an easily displayable HTML page. Client-side scripts (for example, with JavaScript) may not work as intended on older browsers.

3.7.9 Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers. A plain HTML document that the Web daemon **retrieves** is **static**, which means it exists in a constant state: a text file that doesn't change. A CGI program, on the other hand, is **executed** in real-time, so that it can output **dynamic** information.

Since a CGI program is executable, it is basically the equivalent of letting the world run a program on your system, which isn't the safest thing to do. Therefore, there are some security precautions that need to be implemented when it comes to using CGI programs. Probably the one that will affect the typical Web user the most is the fact that CGI programs need to reside in a special directory, so that the Web server knows to execute the program rather than just display it to the browser. A CGI program can be written in any language that allows it to be executed on the system, such as C/C++, Fortran, PERL, TCL, Any Unix shell, Visual Basic and AppleScript. Many people prefer to write CGI scripts instead of programs, since they are easier to debug, modify, and maintain than a typical compiled program.

3.7.10 ASP vs. CGI

Below is the comparison between major options for ASP and CGI:

Criteria	CGI	ASP
OS support	Unix, Dos, Mac, Windows and others.	Windows and Sun Solaris
Scripting Languages	C/C++, Perl, TCL, Fortran, Visual Basic	Vbscript, JavaScript
Learning issues	Not very easy to learn	Easy to learn
Greatest Advantage	Platform Independent and multi-language support.	Rapid application development in a team setting
Limitation	Resource intensive on server if there are multiple requests	Highly IIS dependent, compatibility with other web servers
Costs	Development costs	Development costs
Recommendation	Good for almost any application except when	Good for almost any kind of application, works

	there are going to be large number of hits simultaneously	especially well in Windows NT environment

Table3.2 comparison between major options for CGI and ASP

ASP is chosen in creating the web pages for the following reasons:

- ASP language is six times faster to write than other conventional Web page design methods
- ASP takes less time to write and debug (no compilation), thus less "down" time for Web sites
- ASP outperforms (by 5-to-1 ratio) other conventional Web page design methods (CGI, etc.)
- ASP allows for multiple browsers, doesn't restrict a user to any "one" particular browser type
- ASP Web design is "dynamic", continuous changes can be made "on the fly" effortlessly
- ASP is inherently multi-threaded (CGI isn't) allowing a greater number of concurrent users
- ASP's memory resident engine uses minimal server resources, thus greater performance

3.7.11 OTHER CONSIDERATION

3.7.11.1 Visual Basic

Visual Basic version 6 is the newest addition to the family of Visual Basic products. It use to develop Windows applications quickly and easily for PC without being an expert in C++ or other programming languages. Visual Basic provides a graphical environment in which you visually design the forms and controls that become the building blocks of the applications. Visual Basic supports many useful tools include, but are not limited to, projects, forms, class objects, templates, custom controls, add-ins, and database managers.

Version 6 of Visual Basic is specifically designed to utilize the Internet. It comes with several controls that allow user to create Web-based applications, called *ActiveX executables*. These works just like stand-alone Visual Basic applications, but they are accessed through the Microsoft Internet Explorer 4 Web browser. Using this new style of application, user can revise their existing Visual Basic applications and distribute them through the Internet. New to Visual Basic 6 are the ISAPI Application and Dynamic HTML project templates. These templates provide user with a framework to develop server-side components as well as "smart" Web pages and applications.

3.7.11.2 VBScript

VBScript is a subset of Visual Basic—it will look very familiar to you if you've ever used Visual Basic or Visual Basic for Applications (VBA). It is not identical to these languages, however. Because VBScript was specifically designed to work in browsers, it doesn't include features that are normally outside the scope of scripting, such as file access and printing.

3.7.12 Conclusion

There are several criterias that form the foundation if choosing the appropriate programming languages and technology.

- The language should be able to cater for dynamic Web-based programming.
- The languages should be easily connected to database server.
- The language should be able to provide the implementation of three tier client/server architecture.

Taken as a whole, Microsoft component services include the Component Object Model (COM) and Distributed COM, Microsoft Transaction Server and Microsoft Internet Information Server in Windows NT® Server provide a powerful, flexible, and easy-to-use platform for building distributed applications. It offers the same level of integration, broad tool support, and solid services.

ASP is chosen because it can provides dynamic Web page development and it is tightly integrate with Microsoft Windows NT Server and Microsoft Internet Information Server 3.0. Besides, it can provide various benefits as mentioned in section 3.7.10. ActiveX (which is based on COM/DCOM) is chosen for creation of business object or component (as middle-tier to segregate all the business rules from the server) because it can easily be implemented and the component is also easily be built by using Microsoft Visual Basic 6.0. Microsoft Visual Basic 6.0 is a Rapid Application Development (RAD) programming tools that can help developers to quickly and easily build components. Since ActiveX is mainly based on the COM/DCOM technology, thus COM/DCOM is chosen for development.

Moreover, ActiveX will be expected to be getting more popular in the Windows environment since most of the operating system using in client PC is running Windows 9x. In addition, ActiveX components or controls is optimized for Windows platform and therefore can be executed faster than Java beans. ActiveX provides a standard mechanism to extend any programming language, including Java.

3.8 HARDWARE AND SOFTWARE REQUIREMENTS

3.8.1 Server Side

A minimum of 32 MB of RAM is suggested for a Windows NT Server System, but 64 MB of RAM is more appropriate. It is recommended that implementation of SQL Server in a production environment on anything greater than a Pentium 200. The faster the processor speed and more memory the web server has, the better the performance will be. Network Interface Card (NIC) and network connection with recommended bandwidth at 10Mbps or more, a hard disk at least 5 GB of storage and others standard computer peripherals are also required.

To host and run the system, the server computer needs to have various supporting software installed.

Software/ Components	**Description**
Windows NT Server 4.0	Network Operating System
Internet Information Server	Web-server service
Active Server Pages	Server Scripting Engine
Microsoft SQL Server 7.0	RDBMS for data warehousing
Microsoft Internet Explorer	Precondition for ASP installation
Adobe Acrobat 4.0	Create PDF files from any application

Table 3.3 Server Software Requirements

3.8.2 Client Side

The client hardware requirements are quite minimal as long as it has a reasonable amount of RAM and a reasonable quality dial-up connection line. The recommended configurations are:

1. At least 16 Megabytes of RAM.
2. A minimum 150 MB of hard disk storage.
3. Network connection through existing network configuration.
4. Other standard computer peripherals.

Clients need to have the basic Microsoft Operating System such as Windows 3x or Windows 95 as a basic requirement to install other software to support the system. As for compatibility reason Microsoft products are recommended. The client software requirements fall on the browser used by users. It requires a system that can run IE4 and above or any other browser that support VBScript. Adobe Acrobat Reader 3.01 also required.

Chapter Four

System Design

CHAPTER 4 SYSTEM DESIGN

4.1 INTRODUCTION

System design is a process through which requirements are translated into a model or representation of the software that can be assessed for quality before coding starts. System design is the only way to translate user's requirements into a finished system accurately.

4.2 SYSTEM FUNCTIONALITY DESIGN

4.2.1 Client/Server

Tiers are the physical platforms on which client and server application layers reside. A first generation of client /server applications utilizes two collaborating platforms. The front-end application, residing on an end-user's workstation, provides the application interface and most of the business processing. The back-end application component, residing on one or more hardware processors, provides the database services including data access, management, and integrity services, and some business rule processing in the form of stored procedures and triggers.

Stored procedures are relational DBMS feature that provides processing capability on the server. They have the advantage of centralizing business rules, enforcing security and reducing network traffic by allowing access only to the procedure code, as opposed to the data itself. *Triggers* are stored procedures that can be used to maintain referential integrity and are executed when database is updated by an INSERT, UPDATE or DELETE command.

The three-tier architecture attempts to overcome some of the limitations of the two-tier scheme by separating presentation, functionality logic, and data services into separate distinct entities. In three-tiered architectures, an *application server* is the new third tier and is used to access multiple heterogeneous databases. It generally executes quicker than stored procedures and removes the need to use a proprietary language (like stored procedure SQL). It facilitates reuse of application logic to and allows application code to reside on a variety of platforms. Maintaining business rules is easier instead of redeploying application logic to multiple end-user workstations where application logic reside in a two-tiered architecture, the logic is updated only on the server(s) in which it resides.

4.2.2 Digital Library Theses System

- **Client-tier**

In Digital Library Theses System, the client tier will be the web browser (Internet Explorer 5.0) that display the user interface (web pages) to the user, whereas the application server will be Web Server and the program (or business object written in Visual Basic in DLL format) that serves the web pages to the client and process all request from the user.

- **Middle-tier**

Middle-tier consists of the Internet Information Server 4.0 as the web server. All application programs or files will be resided in the middle-tier (web server). The web server processes the request from the client and then returns required result in the web pages format. It will be process any data request by linking to the database server. For example, authenticating and validating users that login to the system. It is also linked to

Microsoft Transaction Server. For example, when clients are uploading files to the web server.

- **Third-tier**

The third tier consists of the Microsoft SQL Server 7.0 as the database server and Microsoft Transaction Server. Microsoft SQL Server 7.0 maintains the data records. Every query requested from the web server will be authenticated first and the results will then pass back to the web server. Microsoft transaction Server is used to upload files.

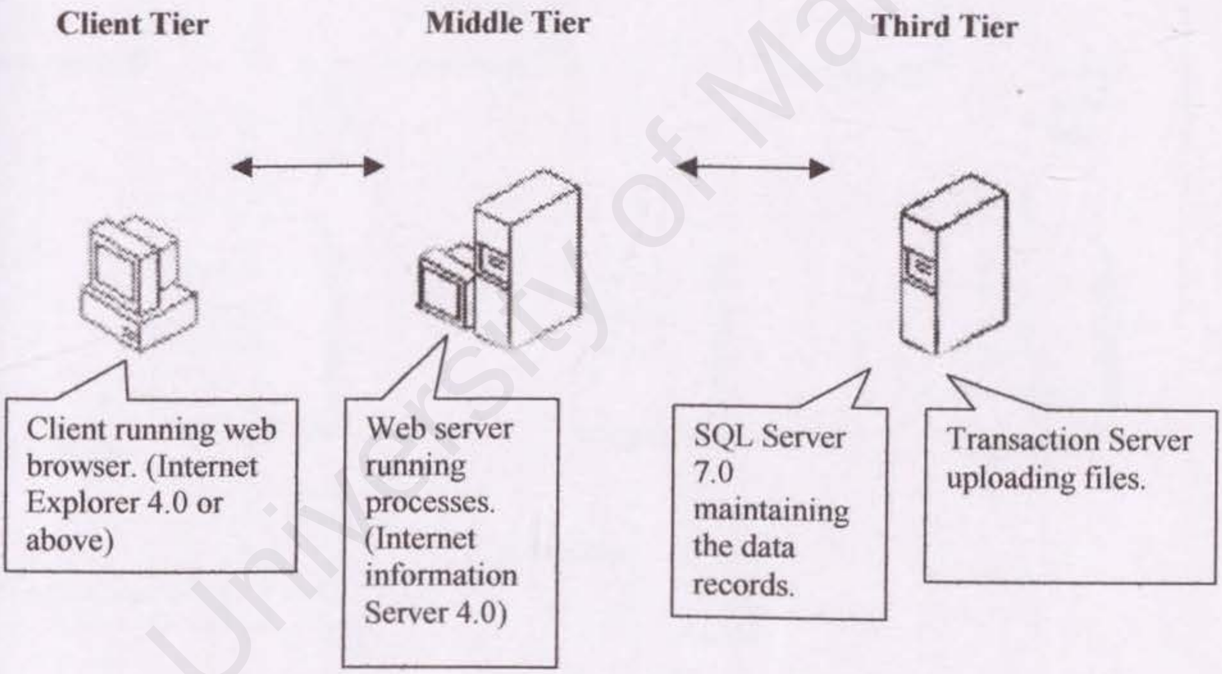

Client Tier
Middle Tier
Third Tier
Client running web browser. (Internet Explorer 4.0 or above)
Web server running processes. (Internet information Server 4.0)
SQL Server 7.0 maintaining the data records.
Transaction Server uploading files.

Figure 4.1 Digital Library System Three-Tier Client Server Architecture

4.2.3 Overall Data Flow Diagram

Data Flow Diagram (DFD) depicts the flow of the data or data to, from and within the system, to storage of data and the transformation function. It also shows the movement of the information logically (what are being processed) and physically (how it is processed). Figure 4.2 represents the data flow diagram for the administrator side of system.

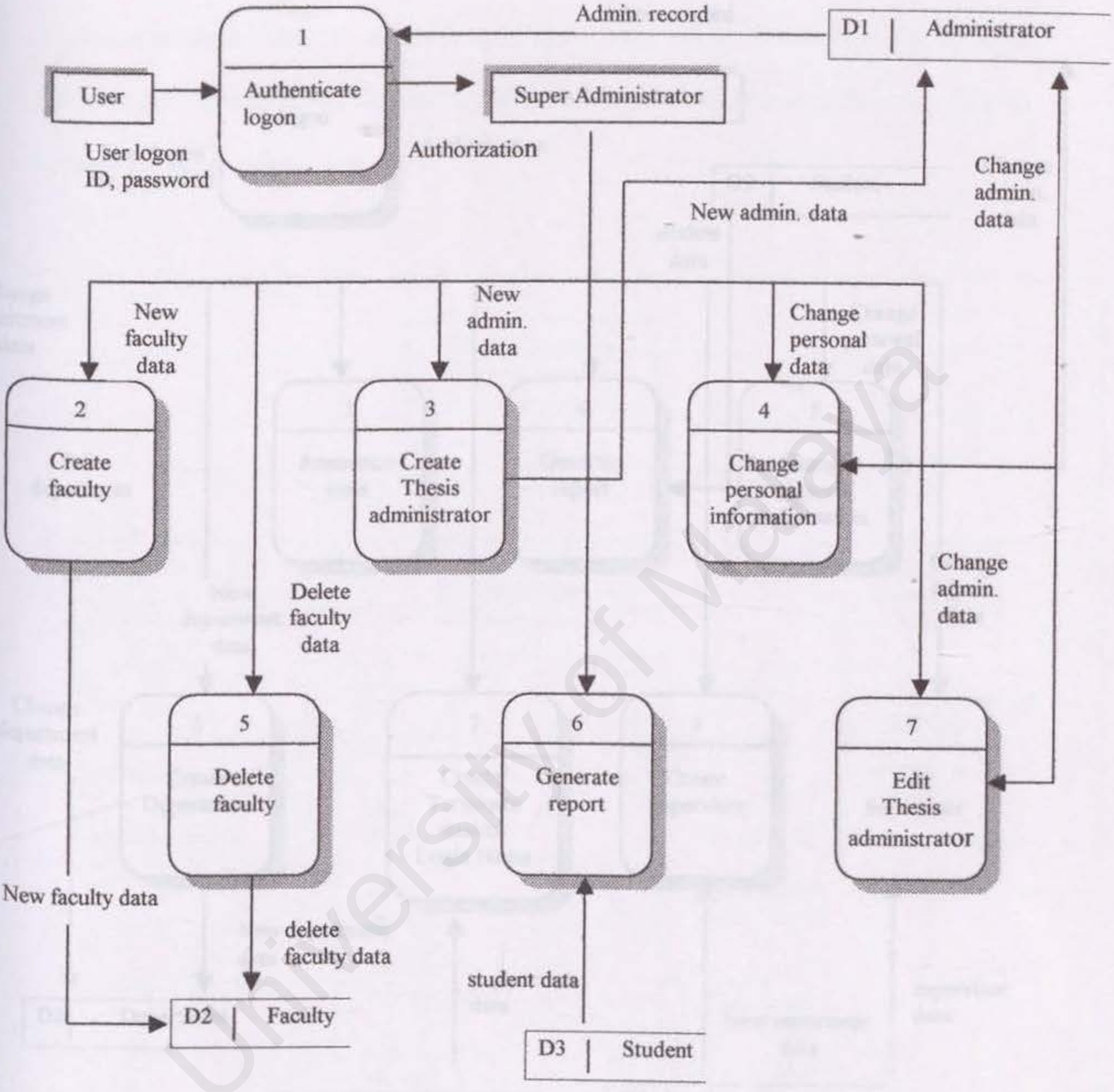


Figure 4.2 Super Administrator's data flow diagram

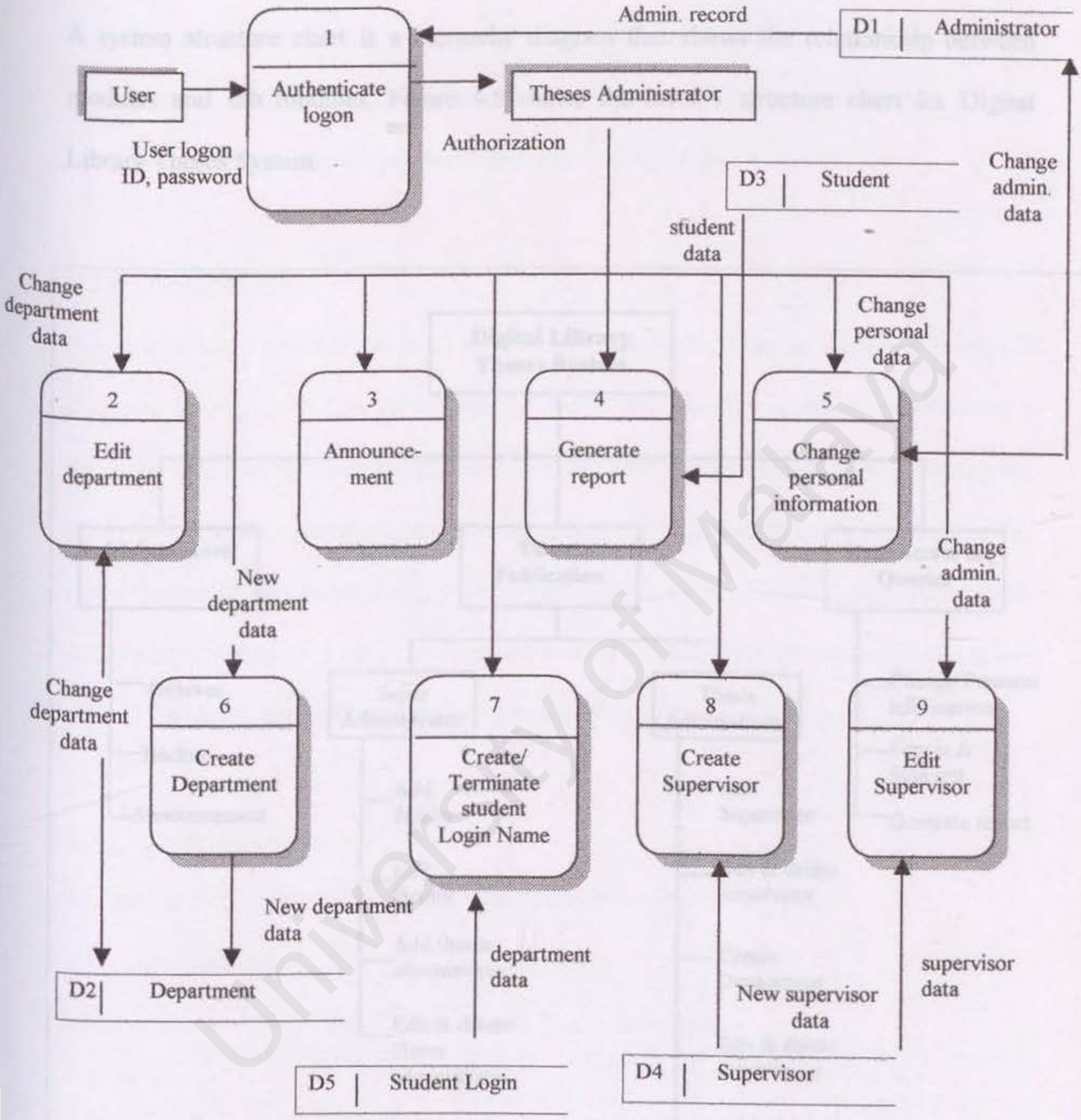


Figure 4.3 Theses Administrator's Data Flow Diagram

4.2.4 System Structure Chart

A system structure chart is a hierarchy diagram that shows the relationship between modules and sub modules. Figure 4.5 shows the level 1 structure chart for Digital Library Theses System.

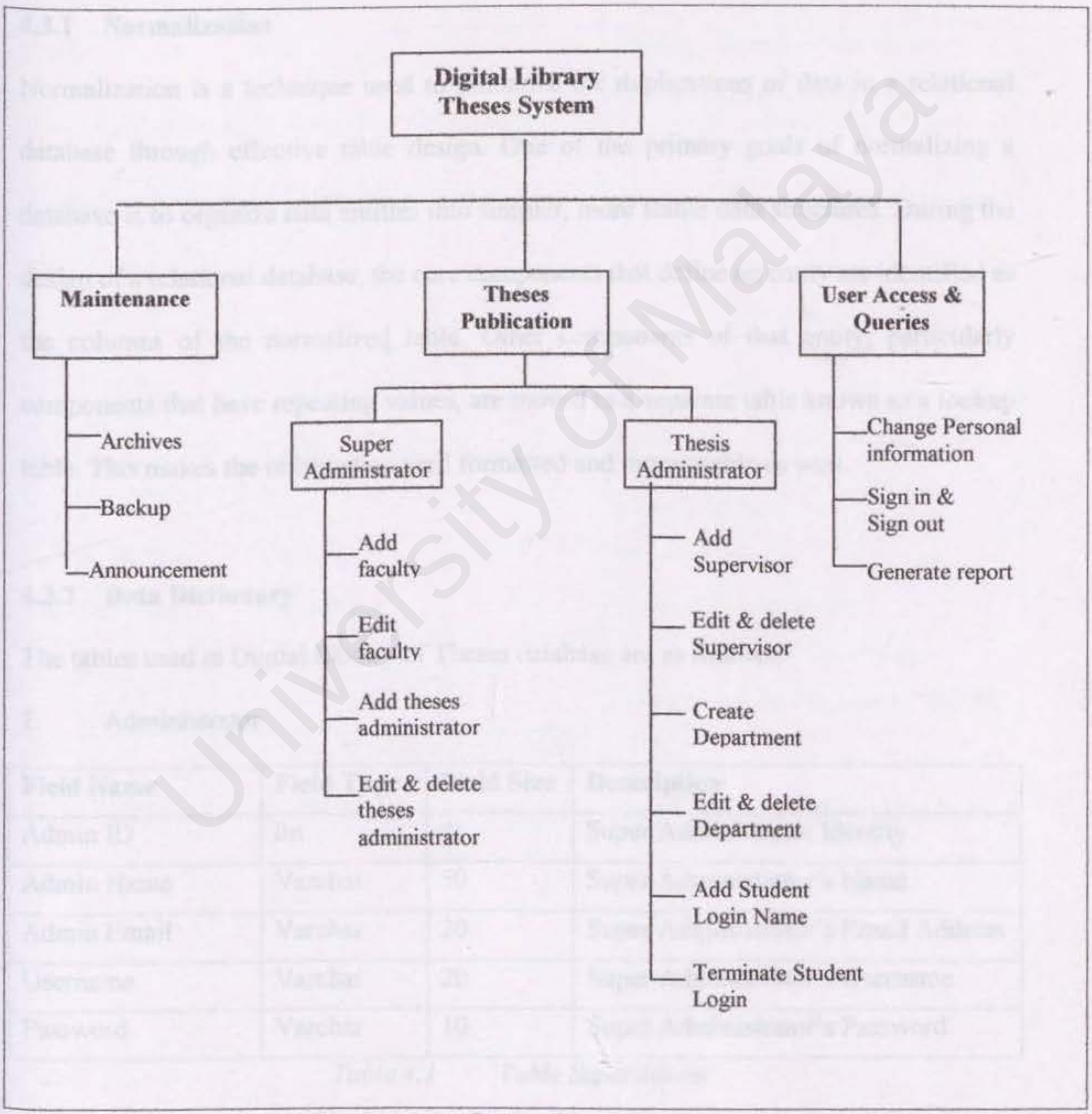


Figure 4.4 System Structure Chart

4.3 DATABASE DESIGN

Database design involves the design of structure used to store and manage data. It transforms the unstructured information and the processing requirements of an application into representations that define the functional specifications.

4.3.1 Normalization

Normalization is a technique used to minimize the duplications of data in a relational database through effective table design. One of the primary goals of normalizing a database is to organize data entities into simpler, more stable data structures. During the design of a relational database, the core components that define an entity are identified as the columns of the normalized table. Other components of that entity, particularly components that have repeating values, are moved to a separate table known as a lookup table. This makes the information well formatted and more usable as well.

4.3.2 Data Dictionary

The tables used in Digital Library of Theses database are as follows:

1. Administrator

Field Name	Field Type	Field Size	Description
Admin ID	Int	4	Super Administrator Identity
Admin Name	Varchar	50	Super Administrator's Name
Admin Email	Varchar	20	Super Administrator's Email Address
Username	Varchar	20	Super Administrator's Username
Password	Varchar	10	Super Administrator's Password

Table 4.1 Table SuperAdmin

Field Name	Field Type	Field Size	Description
Admin ID	Int	4	Thesis Administrator Identity
Admin Name	Varchar	50	Thesis Administrator's Name
Admin Email	Varchar	20	Thesis Administrator's Email Address
Username	Varchar	20	Thesis Administrator's Username
Password	Varchar	10	Thesis Administrator's Password
FacID	Int	4	Faculty Identity which thesis administrator belongs to

Table 4.2 Table ThesisAdmin

Table SUPER ADMIN and THESIS ADMIN store the information of administrator for login purpose.

2. Faculty

Field Name	Field Type	Field Size	Description
FacID	Int	4	Faculty Identity
Faculty Name	Varchar	50	Faculty's Name
Creation Date	Date/time	10	Date of faculty create
File	Varchar	20	Physical path of file which contains the description of faculty

Table 4.3 Table Faculty

Table FACULTY stores the faculty information for 'home' page and reference as lookup table.

3. Department

Field Name	Field Type	Field Size	Description
DeptID	Int	4	Department Identity
Department Name	Varchar	50	Department's Name
Creation Date	Date/time	10	Date of faculty create
FacID	Int	4	Faculty Identity which department belongs to

Table 4.4 Table Department

Table DEPARTMENT stores the department information in a faculty to enable the listing of theses and reference as lookup table.

4. Supervisor

Field Name	Field Type	Field Size	Description
SupervisorID	Int	4	Supervisor Identity
Supervisor Name	Varchar	50	Supervisor's Name
FacID	Int	4	Faculty Identity which supervisor belongs to

Table 4.5 Table Supervisor

Table SUPERVISOR uses as a lookup table to store the supervisor name so that the database is in third normal form.

5. Student Login

Field Name	Field Type	Field Size	Description
Username	Varchar	20	Student's username for login of submission form
Password	Varchar	10	Student's password for login of submission form
Email	Varchar	20	Student's Email Address

Table 4.6 Table Login Student

Table LOGIN STUDENT stores the username and password of student, which will be used to login in submission form of thesis later. After certain period the records in table will be deleted to terminate student from login. After the username and password created, an email will be sent to that student immediately.

6. Announcement

Field Name	Field Type	Field Size	Description
Creator	Varchar	50	Name of announcement's creator
Target	Varchar	50	Announcement's target
Creation Date	Date/time	10	Date of faculty create
Content	Varchar	700	Content of the announcement

Table 4.7 Table Announcement

Table ANNOUNCEMENT stores the announcement that will be publish to the web page on client-side.

4.4 USER INTERFACE DESIGN

The goal of interface design is to provide the best way for people to interact with computers. Provision of good interfaces is very important because the impact to the organizations is increasing. Most people in organizations are spending more time interacting with computers as part of their work. Their performance and effectiveness of the organizations will be improved with better interfaces.

Nowadays, people are more interested of having a too] that is simple and easy to use instead of learning the technology behind the computer. They don't want to spend time learning on the software, they likes software that is user-friendly. With a good user interface, users will no longer need to remember a lot of commands. Hence, it will improve the efficiency and the effectiveness of user using the system.

4.3.1 Screen Design

All user interfaces in this system are mainly created by using Markup Language (HTML), Adobe Photoshop, Active Server Page and VB Script.

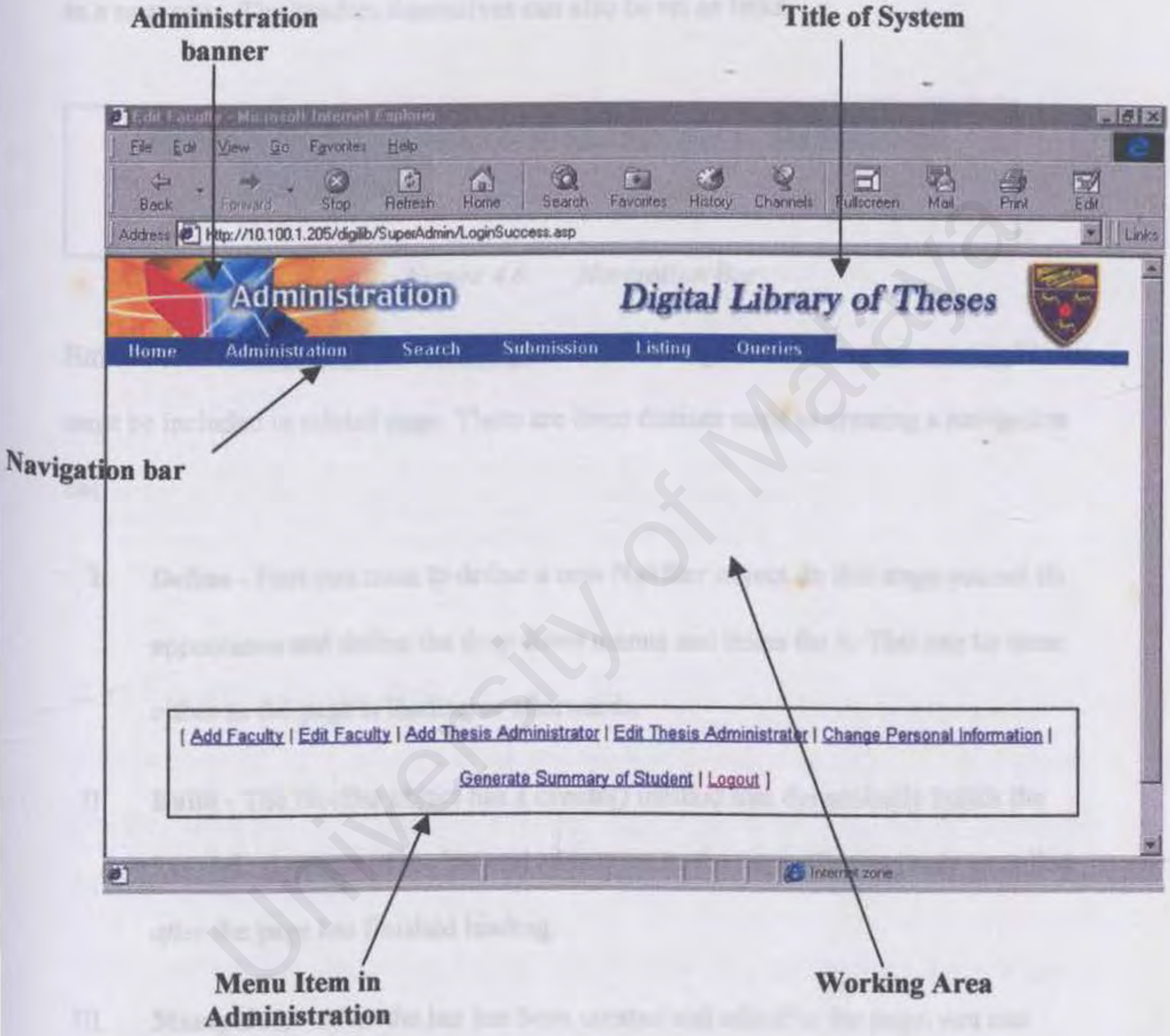

Administration banner
Title of System
Edit Faculty - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address http://10.100.1.205/digilib/SuperAdmin/LoginSuccess.asp
Links
Administration
Digital Library of Theses
Home Administration Search Submission Listing Queries
Navigation bar
[Add Faculty | Edit Faculty | Add Thesis Administrator | Edit Thesis Administrator | Change Personal Information |
Generate Summary of Student | Logout]
Internet zone
Menu Item in Administration
Working Area

Figure 4.5 Interface design

4.3.2 Navigation Bar

Mouse over the headers at the top of the page and you'll get a drop down menu. The individual menu items are highlighted when under the cursor and can be clicked to link to a new page. The headers themselves can also be set as links.

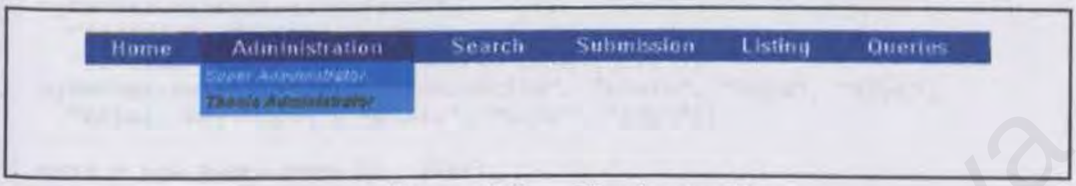


Figure 4.6 Navigation Bar

Either both the dhtmllib.js and navbar.js files or the single, condensed navcond.js file must be included in related page. There are three distinct steps to creating a navigation bar.

I. **Define** - First you must to define a new **NavBar** object. In this stage you set its appearance and define the drop down menus and items for it. This can be done either as the page is loading or afterwards.

II. **Build** - The NavBar object has a **create()** method that dynamically builds the DHTML elements of the bar and adds them to the page. This can only be called *after* the page has finished loading.

III. **Manipulate** - After the bar has been created and added to the page, you can position it, hide or show it or change some of its attributes like width, alignment and stacking order.

I. Defining a Navigation Bar

A navigation bar is made up of three distinct objects: one for the bar itself, one for drop down menus and one for individual items. You use these objects and their associated methods to construct a bar. Here is some sample code.

```
var myNavBar, menu;

myNavBar = new NavBar(500);
myNavBar.setSizes(1, 2, 1);
myNavBar.setColors("#000000",
  "#ffffff", "#669999", "#000000", "#66cccc",
  "#ffffff", "#339999", "#000000", "#99ffff");
myNavBar.setFonts("Arial, Helvetica", "plain", "bold", "10pt",
  "Arial, Helvetica", "plain", "bold", "10pt");

menu = new NavBarMenu(80, 100);
menu.addItem(new NavBarMenuItem("Header 1", "blank.html"));
menu.addItem(new NavBarMenuItem("Item 1-1", "blank.html"));
menu.addItem(new NavBarMenuItem("Item 1-2", "blank.html"));
menu.addItem(new NavBarMenuItem("Item 1-3", "blank.html"));
menu.addItem(new NavBarMenuItem("Item 1-4", "blank.html"));
myNavBar.addMenu(menu);

menu = new NavBarMenu(80, 100);
menu.addItem(new NavBarMenuItem("Header 2", "blank.html"));
menu.addItem(new NavBarMenuItem("Item 2-1", "blank.html"));
menu.addItem(new NavBarMenuItem("Item 2-2", "blank.html"));
menu.addItem(new NavBarMenuItem("Item 2-3", "blank.html"));
myNavBar.addMenu(menu);

...
```

II. Building a Navigation Bar

Once you've defined a navigation bar and its menus you need to build the necessary DHTML elements for it and add it to the page. This is done by simply calling the NavBar create() method.

You can only call **create()** *after* the page has finished loading. Once created, you cannot change the colors or fonts of the bar or add new menus or items to it.

The easiest way to do this it so set up a function to call via the onload event in the <BODY> tag. Here is a typical page setup.

```
<html>
<head>
<title></title>
<script language="JavaScript" src="dhtmllib.js"></script>
<script language="JavaScript" src="navbar.js"></script>
<script language="JavaScript">
```

```
// Define the navigation bars.

var myNavBar1 = new NavBar(400);
var myNavBar2 = new NavBar(0);

/* ... see code above for configuring the navigation bars ... */

function init() {

  // Create the navigation bars.

  myNavBar1.create();
  myNavBar2.create();
}

</script>
</head>
<body onload="init()">

<!-- your page contents -->

</body>
</html>
```

III. Manipulating a Navigation Bar

Once the bar has been added to the page, you can reposition it, resize it, hide and show it, etc. as desired. Frames are treated as individual windows by the browser. The bar and drop down menus cannot cross over the boundaries between frames, any more than an image on a page can be made to stick outside of the browser window. Likewise, you cannot have a drop down appear in a frame other than the one containing its parent bar.

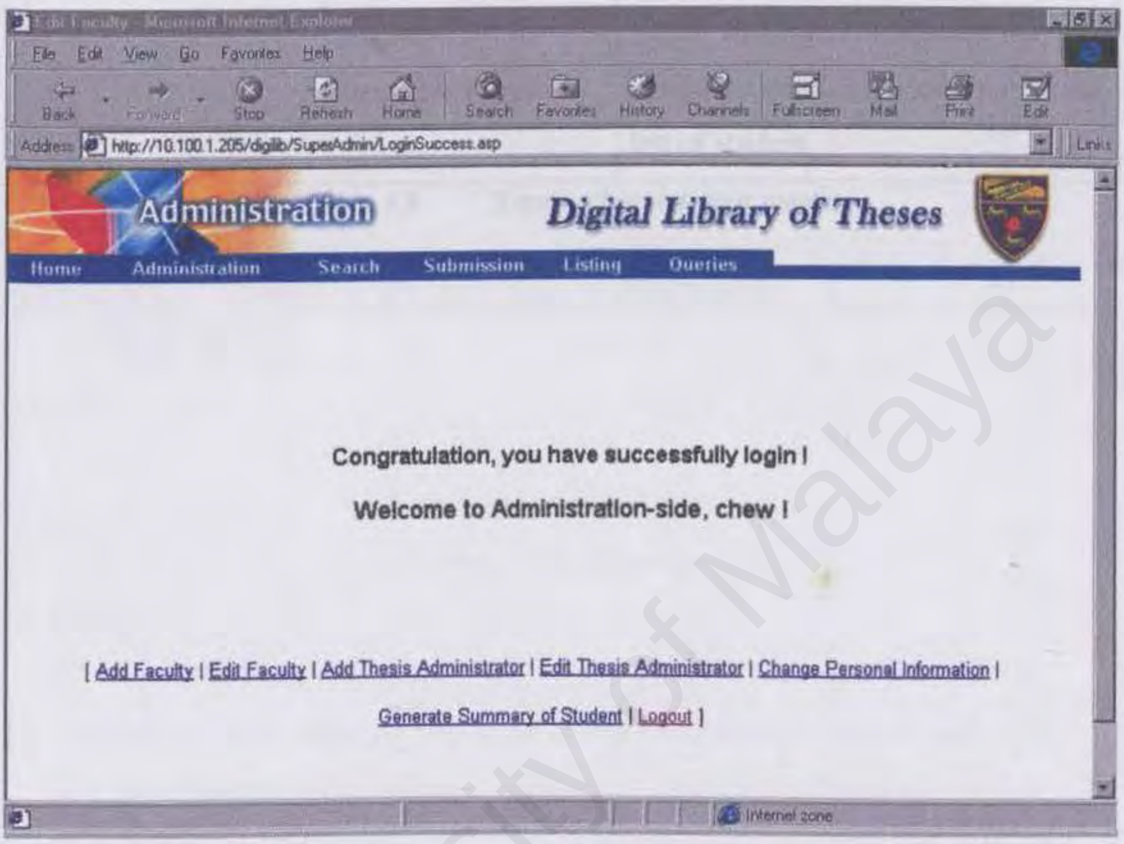
The bars and menus are dynamically positioned objects which float on top of the normal page contents. That means that if you want one to appear in a particular location on your page you'll have to find the coordinates of that spot and use the **moveTo()** method to place it there.

4.3.3 Design of Menu

Since each type of administrator will have the different authorization, the menu that displayed out will be different. The authorization of each administrator is recognized by the system through log in process according to the login ID.

4.3.3.1 Super Administrator

Figure 4.7 is the menu for super administrator and table 4.8 is the description.


Edit Faculty - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address
Links
Administration
Digital Library of Theses
Home Administration Search Submission Listing Queries
Internet zone

http://10.100.1.205/digilib/SuperAdmin/LoginSuccess.asp

Congratulation, you have successfully login !

Welcome to Administration-side, chew !

[Add Faculty | Edit Faculty | Add Thesis Administrator | Edit Thesis Administrator | Change Personal Information | Generate Summary of Student | Logout]

Figure 4.7 Super Administrator Menu

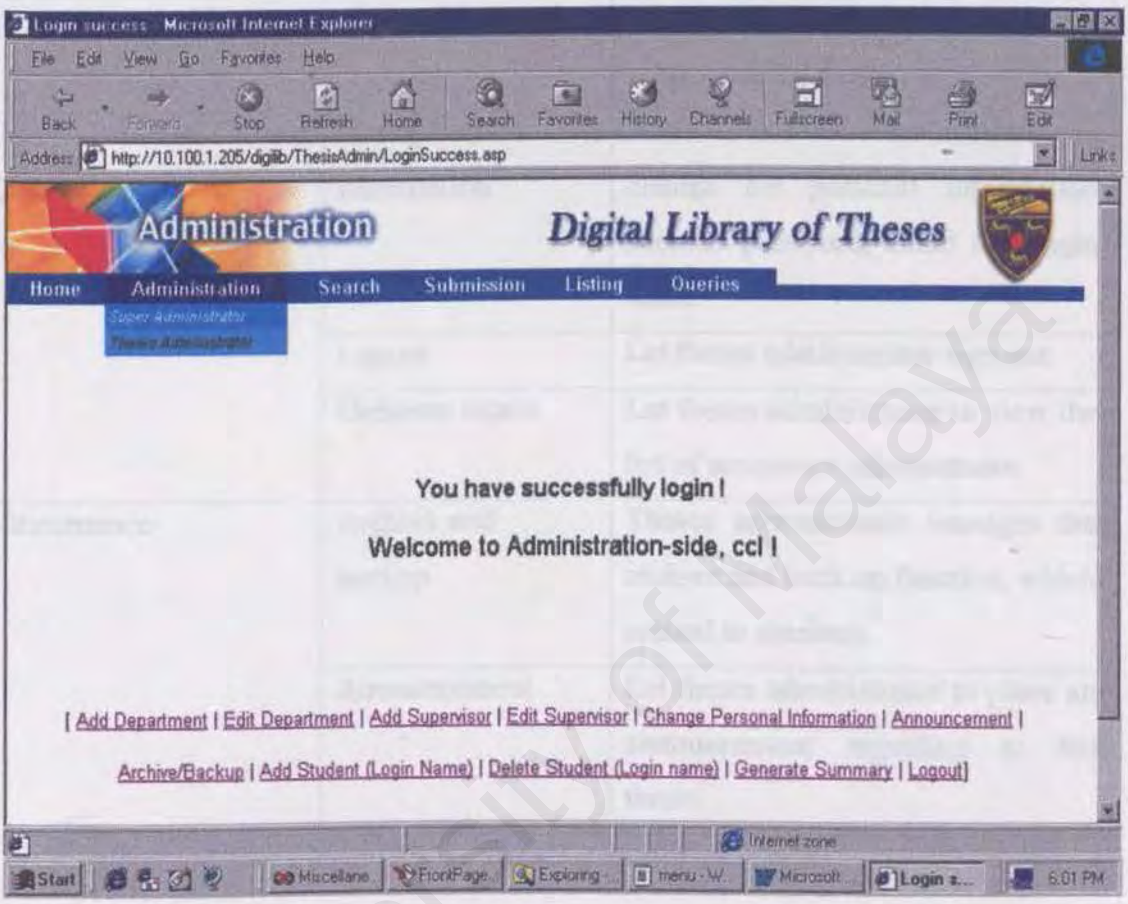
Module	Menu Items	Description
Theses Publication	Add faculty	Let super administrator add new joining faculty.
	Edit faculty	Let super administrator edit existing joining faculty.
	Add theses administrator	Let super administrator add new theses administrator.
	Edit & delete theses administrator	Let super administrator edit or delete theses administrator.

User Access and Queries	Change my information	Assist super administrator to change his personal information such as password, email and login name.
	Logout	Let super administrator sign out.
	Generate report	Let super administration to view the list of student.

Table 4.8 Super administrator menu

4.3.3.2 Theses Administrator

Figure 4.8 is the menu for Theses administrator and table 4.9 is the description



http://10.100.1.205/digilib/ThesisAdmin/LoginSuccess.asp

You have successfully login !

Welcome to Administration-side, ccl !

[Add Department | Edit Department | Add Supervisor | Edit Supervisor | Change Personal Information | Announcement | Archive/Backup | Add Student (Login Name) | Delete Student (Login name) | Generate Summary | Logout]

Figure 4.8 Theses Administrator Menu

Module	Menu Items	Description
Theses Publication	Add department	Let theses administrator to add new department.
	Edit and delete department	Let theses administrator edit existing department.
	Add supervisor	Let theses administrator add new supervisor.
	Edit & delete supervisor	Let theses administrator edit or delete supervisor.

	Add Student (login name list)	Let theses administrator add new student login name.
	Terminate Student login	Let theses administrator delete student login name to terminate the submission of theses.
User Access and Queries	Change my information	Assist theses administrator to change his personal information such as password, email and login name.
	Logout	Let theses administrator sign out.
	Generate report	Let theses administrator to view the list of assistance administrator.
Maintenance	Archive and backup	Theses administrator manages the archive and back up function, which related to database.
	Announcement	Let theses administrator to place an announcement regarding to the thesis.

Table 4.9 Theses administrator menu

Chapter Five

System Implementation

CHAPTER 5 SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

System implementation involves the translation of the software representation produced by the design phase into a computer-readable form. System implementation's major tasks include coding, testing and debugging. The coding phase will invōlving program the new system application and testing phase involves test on application that developed while debugging phase involves fixing the bug occurred in the program.

5.2 CODING

Coding is a process that translates a detail design representation of software into a programming language realization. It translates the design representation into a machine-readable form. Microsoft Visual InterDev 6.0 is used in creating all modules. In the process of coding, every algorithm that is designed during designing is transformed into lines of codes.

5.2.1 Coding Methodology

The coding methodology used in the development of Digital Library of Theses is the top-down with a little bottom-up approach.

Top-down Approach

This is a design method in which the modules to be accomplished is broken down into sub modules and each of the submodules is then further decomposed into smaller submodules, and so forth. This is often a good way of constructing a program eventually the sub-problems is small enough that can be written in functions. The top-down approach will ensure the most important modules will be developed and tested first. By using this approach, the administration of Digital Library system is divided into several modules, namely maintenance, thesis publication and user access and queries. Each of these modules is then further divided into smaller submodule.

Bottom-up Approach

Bottom-up design starts with the coding of the lower level modules before the higher level modules are constructed. The higher modules are just the skeletons that call the lower modules.

5.2.2 Administration Side Coding

Since each type of administrator has different authorization, the access right of each administrator will be checked at the beginning of each web page before they are allowed to do further processing. For example, refer the coding below.

5.2.2.1 Maintenance

Filename	Description/ Main Function
NewAnnouncement.asp	Create a new announcement.
PostAnnounce.asp	Insert new announcement into database.
Announcement.asp	Display a new announcement on client side.
Home.asp	First page of faculty which display the faculty name and some short description.

Table 5.1 Administration-side maintenance coding

5.2.2.2 Thesis Publication

a) Super Administrator

Filename	**Description/ Main Function**
NewFaculty.asp	Provide form to create new faculty.
Fac.asp	Add new faculty information into database and file.
EditFaculty.asp	Provide form to change faculty's information.
ModifyFaculty.asp	Display the selected faculty's information.
Updatefac.asp	Update or delete the faculty.
NewThesisAdmin.asp	Provide form to create thesis administrator's information.
AddTAdmin.asp	Add new thesis administrator's information into database and file.
EditThesisAdmin.asp	Provide form to change thesis administrator's information.
ModifyTAdmin.asp	Display the selected thesis administrator's information.
UpdateTAdmin.asp	Update or delete thesis administrator's information.
StudentSum.asp	Display a list of student's metadata.

Table 5.2 Administration-side thesis publication (super administrator) coding

b) Thesis Administrator

Filename	**Description/ Main Function**
NewDepartment.asp	Provide form to create new department.
Adddept.asp	Add new department information into database and file.
EditDepartment.asp	Provide form to change department's information.
ModifyDepartment.asp	Display the selected department's information.
UpdateDept.asp	Update or delete the department.
NewSupervisor.asp	Provide form to create supervisor's information.
AddSupervisor.asp	Add new supervisor's information into database and file.
EditSupervisor.asp	Provide form to change supervisor's information.
ModifySupervisor.asp	Display the selected supervisor's information.
UpdateSupervisor.asp	Update or delete supervisor's information.
StudentList.asp	Display a list of student's metadata.
NewStudent(login).asp	Provide form to add new student login name.
AddStudent(login).asp	Add new supervisor's information into database and file.
TerminateStudent(login).asp	Provide form to delete student's records.
Deletestudent(login).asp	Delete student's information.
ChkStatusTAdmin.asp	Check authorization of thesis admin.

Table 5.3 Administration-side thesis publication (thesis administrator) coding

5.2.2.3 User Access and Queries

Filename	Description/ Main Function
Login_SAdmin.asp	Provide form for Super Administrator to login.
Login_TAdmin.asp	Provide form for Thesis Administrator to login.
ChkloginSAdmin.asp	Process the login of Super Administrator.
ChkloginTAdmin.asp	Process the login of Thesis Administrator.
Logout.asp	Display message that administrator has successfully logout and close all the session.
Loginsuccess.asp	Display message that administrator has successfully login.
AdminDetailSAdmin.asp	Provide form to change super administrator's information.
ChangeSAdminDetail.asp	Update super admin information.
AdminDetailTAdmin.asp	Provide form to change Thesis administrator's information.
ChangeTAdminDetail.asp	Update thesis admin information.

Table 5.4 Administration-side user access & queries coding

5.3 TESTING

Testing is the process of determining whether a program or system performs the desired processing. Testing cannot show the absence of defects, it can only show that software defects are present. Although testing is tedious, it is an essential series of steps that helps assure the quality of the eventual system. It is far less disruptive to test beforehand than to have a poorly tested system failed after implementation. In developing this system, several stages are used for testing. First, each program component is tested on its own, isolated from the other components in the system. It is called as unit testing. It is followed by integration testing. During integration testing, modules are integrated with the interface built, and tested together. It is the process of verifying that the system components work together as described in the system. Next, interface testing is carried out and the whole system is tested.

5.3.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design. The modules are tested independently of one another to ensure their correct operation. In Digital Library of Theses, modules that make up the system are mostly form modules which perform a specific function. Each of these form modules contains many subfunctions. For instance, the function to maintain supervisor in the system contains subfunctions like add supervisor, delete supervisor, update supervisor and so forth. Therefore, unit testing was conducted oil each subfunction first and followed by the module itself. Each unit testing is using 2 methods, white box testing and black box testing.

White box testing consists of six types code coverage that listed as below:

1. **Segment coverage**

   Every segment of the code between control structures is supposed to execute at least once.

2. **Branch node coverage**

   Every branch at every possible direction is taken at least once.

3. **Compound condition coverage**

   When multiple conditions appear in the code, every possible combination is tested based on a truth table.

4. Data flow testing

Data flow testing is meant for reflecting dependencies which are mainly caused by sequences of data manipulations.

5. Loop testing

This type of testing is related to looping. Loop testing is difficult to test when dependencies exist among the loops or between a loop and the code it contains.

6. Basic path and path testing

Each independent path throughout the code is usually taken at a predetermined order. When dependencies appears in the code, each path where dependency appears exists must be tested.

Black box testing includes error guessing and boundary value analysis. This type of testing is used to find incorrect or mission functions, interface errors, errors in data structures, performance errors and initialization and termination errors. Tester will deliberately do some mistakes or enter values that are out of acceptable range to test the system reaction and to determine whether the system will prompt out the appropriate error. Besides, It also tests the functionality of the system in an ad hoc basis without knowing the logic structure of the code. Input is provided and output is verified manually to check for accuracy.

5.3.2 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. Integration testing is used to verify the combining of the software units. It addresses the issues associated with the dual problems of verification and program construction.

5.3.3 System Testing

System testing is actually a series of different test to fully exercise the system. Although each test has different purpose, all work should verify all system elements (e.g. Software, hardware and interface etc.) are fully functioning and integrate properly.

5.4 SOFTWARE TOOLS

Certain software tools have been used to develop the system. Those help us to build up the system quickly and easily.

5.4.1 Development Tools

Table 5.5 lists the software tools used for system development.

Software	Description
Microsoft Visual Interdev 1.0	Visual tool for building dynamic web application using Active Server Pages. It also helps in placing the ActiveX control into the web pages.
Microsoft SQL Server 7.0	Database server used for storing and manipulating vast amount of data. It contains all relational tables, database structure, data, stored procedures and task scheduling.
Microsoft Access 2000	Database management system used for porting and building all the relational tables, structures and data. Access 2000 is using for trying purpose before put into SQL Server for storage and manipulation because it provides a visual environment in dealing with database.

Table 5.5 System development tools

5.4.2 User Interfaces Design Tools

Table 5.6 lists the software tools used for interfaces design.

Software	Description
Microsoft FrontPage 2000	Web authoring tool for building web interface design. It also helps in placing the form components into the web pages.
Adobe Photoshop 4.0	Graphical tool used for creating and editing images.

Table 5.6 Description of graphical tools

5.4.3 Documentation Tools

Table 5.7 lists the software tools used for documentation.

Software	Description
Microsoft Word 2000	Word processor for writing documentation of thesis.
Microsoft PowerPoint 2000	Tool used to create slides for presentation.
Microsoft Project 98	Project management tool for creating project schedule in GANTT chart format.

Table 5.7 Description of documentation tools

Chapter Six

Evaluation and Conclusion

CHAPTER 6 EVALUATION AND CONCLUSION

6.1 PROBLEMS AND SOLUTIONS

In the process of developing the system, various problems have been arose which some have overcome and some of the problems are yet to discover and overcome. The major problems and the solutions taken to overcome them are explained in the following sections.

6.1.1 Lack of knowledge and experience in web-based programming

The web application development is quite complex compare to the traditional stand-alone application. The programming of web-based application does not only involve the setting of the server and the development of executable codes, structure designing is also required. In order to develop the system, research and study on related material have been carried out. Much reading and analysis have provided assistance in solving various problems. Besides, discussing with friends and participating in web-side forums are also helpful in order to solve the problems.

6.1.2 Lack of knowledge and experience in interface designing

Interface designing requires not only the graphical tools such as Photoshop but also sense of art. Due to lack of graphic design knowledge, several interface functions and design under planning are unable to be developed. These functions are suggested in the future enhancement part.

6.1.3 Lack of knowledge and experience in database server

Dealing with database server to store and manipulate the entire data of system came across a lot of problem. The method of administer and manipulate the database is rather different as compare to non-database server program. It consists of a lot of new features such as stored procedures, trigger and task scheduling. Therefore, a lot of time has been taken in learning the database server concept. Surfing through Internet, reading up the related material and discussing with friends did help in overcome the problem.

6.2 SYSTEM STRENGTHS

6.2.1 Friendly User Interface

Digital Library of Theses provides an attractive and ease of use interface. User can navigate the system using the drop down navigation bar by "point and click" method using mouse. Therefore, the time for user to learn to use the system and to train new user has been reduced.

6.2.2 Time-effectively with Less Manpower Theses system

Digital Library of Theses helps administrators to manage theses time-effectively with less manpower. Administrators may manage theses on-line and do not need to go to the web server remotely. For example, administrators may change personal information on-line to the web server. E-mail will be sent to student when the registration of login name by administrator is complete. In addition, administrators are divided into different group with different authorization. For example, super administrator can create thesis administrators to assists them.

6.2.3 Error Controlling

When a process is unsuccessfully done, a user-friendly error message will be alerted. For example, when an administrator inserts duplication records into a table, a message with *"Records already exist"* will be display on the form. This system also tries to decrease the total number of errors. Faculty ID, Department ID, Administrator ID and other primary keys in tables are auto generated by the system.

6.3 SYSTEM LIMITATIONS

6.3.1 Browser Limitation

The system only limited to certain browsers. Most of the scripts of this system are written in VBScript, which is not supported by other browsers such as Netscape Communication. Thus, it can only run with browser Internet Explorer 4.0 or above.

6.3.2 No Context Sensitive Help

Although the system does provide searchable on-line help for user, it doesn't provide context sensitive help. If users getting some difficulty in using a specific function or module, they still need to search the topic of the entire help.

6.3.3 No Encryption for password

Password field in tables are not encrypted and therefore it may exposes the administrator's password to any malicious user. Furthermore, the system is not using secure connection, Secure Sockets Layer (SSL) to protect the sensitive information.

6.3.4 Unsupport dual lingual

Digital library of theses does support English (Roman) character only for bibliographic data. Other lingual such as Chinese and Malay character are not supported by the system.

6.4 FUTURE ENHANCEMENT

6.4.1 Security

Password field in tables should be encrypted to avoid the exposure of applicant's password to any malicious user. Password that sent on-line must be encrypted to ensure the security and the confidentiality of the system information.

6.4.2 Animated User Interface

Digital Library of Theses could be including some animated user interface to make the site more attractive. However, not too many animated picture should be put in to avoid the site become too fancy and confusing.

6.4.3 Online theses system

The Digital Library of Theses system should go to Internet and not limited to Intranet only. A few new features can be added if the system goes online such as subscription, online payment for theses and etc.

6.4.4 Subscription and Online payment

If the system goes online, this function is very important so that interested people can be subscribed to Digital Library of Theses. However, if the Internet users want to get the full text of thesis, then he or she should pay for it. It is more convenient if users can pay the subscription fee on-line. After a payment is done, administrator should be notified and a weekly report should be generated.

6.4.5 Editorial Board

If the system goes online, then there should be an editorial board, which can control the qualities of theses that put on the Internet. Besides, the committee of Editorial Board can edit the abstract of thesis for better illustration.

6.4.6 Mail server

Mail server enable administrator to send and reply mail to the user of the system. It also allows administrator to create email account to any other users. For an example, administrator will by notified by email after student submit the thesis through mail server.

6.4.7 Secure Socket Layer

If the system goes online, secure sockets layer will ensure that sensitive information passing through an Internet link (TCP/IP connection) is safe from prying eyes. When both the client and server support SSL, any data transmitted between the two becomes encrypted.

6.4.8 Dual lingual support

The system should not provide only English but also Malay character support. The users can even search the bibliographic data using both of the languages. Moreover, the system can have a simple lookup table for referring English words to Malay words.

6.5 CONCLUSION

As a conclusion, this project has met its objective of developing a web-based Digital Library of Theses system. The implementation of web-based theses system will sure help administrator to reduce their workloads. It will also ease the thesis management and at the same time promote a paperless environment. It also offers a plaçe to promote exchange of information and knowledge in research work, new inventions and developments effectiveness.

A lot of valuable knowledge was gained throughout the development of this project. It has provided a golden opportunity to learn the fundamental of web-based programming language in ASP and VBScript besides the knowledge and experience in setting, configuring and manage the Windows NT Server, Web Server (IIS) and SQL Server.

The system will become more powerful and complete as the new enhancements and features such as online theses system, subscription and online payments, editorial board, secure sockets layers, mail server and dual lingual support, add on to it in the near future.

Bibliography

BIBLIOGRAPHY

[illegible] Architecture OMG, July 1995
http:// www.cs.[illegible]/CORBA-docs/corba2.0.ps.gz

Steve Vinoski, CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments, *IEEE Communications Magazine*, February, 1997.
http:// www.cs.wustl.edu/~schmidt/vinoski.ps.gz

Matthew G. Kirschenbaum (1996). "Electronic publishing and doctoral dissertations in the humanities." 1996 Annual Convention of the Modern Language Association, Washington DC; see also a revised and expanded version at Joint Annual Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, Kingston, Ontario, June 6, 1997.
http://etext.lib.virginia.edu/ETD/about/etd-mla.html

Matthew G. Kirschenbaum (1998). "Electronic theses and dissertations in the humanities: A directory of on-line references and resources."
http://etext.lib.virginia.edu/ETD/ETD.html

Katherine S. Mangan (1996). "Universities consider whether new format is appropriate way to present research" *Chronicle of Higher Education*, March 8, 1996. Page A 15.
http://etext.lib.virginia.edu/ETD/about/chronicle.html

DCOM Technical Overview, November 1997
http:// www.microsoft.com/ntserver/library/dcomtec.asp

Java Language Overview, December 1997
http://www.javasoft.com/docs/Overviews/java-overview-1.html

Kenneth E. Kendall and Julie E. Kendall, "*System Analysis and Design*", 3rd Edition, Prentice Hall International, Inc.,1995

BIBLIOGRAPHY

Common Object Request Broker Archictecture, OMG, July, 1995.
http:// www.cs.wustl.edu/~schmidt/CORBA-docs/corba2.0.ps.gz

Steve Vinoski, CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments, *IEEE Communications Magazine*, February, 1997.
http:// www.cs.wustl.edu/~schmidt/vinoski.ps.gz

Matthew G. Kirschenbaum (1996). "Electronic publishing and doctoral dissertations in the humanities." 1996 Annual Convention of the Modern Language Association, Washington DC; see also a revised and expanded version in Proc. Joint Annual Conference of the Association for Computers and the Humanities and the Association for Literary and Linguistic Computing, Kingston, Ontario, June 6, 1997. http://etext.lib.virginia.edu/ETD/about/etd-mla.html

Matthew G. Kirschenbaum (1998). "Electronic theses and dissertations in the humanities: A directory of on-line references and resources." http://etext.lib.virginia.edu/ETD/ETD.html

Katherine S. Mangan (1998). "Universities consider whether new format is appropriate way to present research." *Chronicle of Higher Education*, March 8, 1996. Page A 15. http://etext.lib.virginia.edu/ETD/about/chronicle.html

DCOM Technical Overview, November 1997
http:// www.microsoft.com/ntserver/library/decomtec.exe

Java Language Overview, December 1997
http://www.javasoft.com/docs.Overview/java-overview-1.html

Kenneth E. Kendall and Julie E. Kendall, "*System Analysis and Design*", 3rd Edition, Prentice Hall International, Inc.,1995

Ian Sommerville, "*Software Engineering*, 5th Edition", Addition Wesley, 1996

Elliotte Rusty Harold, Java Developer's Resource, Prentice Hall PTR Prentice Hall, Inc.,1997

Network Digital Library of Theses and Dissertations,
http://www.ndltd.org,
http://www.dlib.org/dlib/september97/theses/09fox.html,
http://www.theses.org,
http://www.ndltd.org/related/project.htm

Virginia Tech,
http://etd.vt.edu/
http://etd.edu/howto/index.htm
http://scholar.lib.vt.edu/theses/archive.html

West Virginia University
http://www.wvu.edu/~theses/

University of Waterloo,
http://www.and.uwaterloo.ca, http://www.lib.uwaterloo.ca

University of Toronto,
http://www.fis.utoronto.ca/etd

UMI,
http://www.umi.com/

Pockley, Simon, "Killing the Duck to Keep the Quack,"
http://www.cinemedia.net/FOD/FOD0055.html

Exon,Maggie, "Long-Term Management Issues in the Preservation of Electronic Information,"
http://www.nla.gov.au/niac/meetings/npo95me.html, Nov. 1995

"Preserving Digital Information Report of the Task Force on Archiving of Digital Information", commissioned by the Commission on Preservation and Access and the Research Libraries Group, Inc.,
http://lyra.rlg.org/ArchTF/tfadi.index.htm

Active Server Pages,
http://www.aspin.com/index/default.asp

Java Beans,
http://java.sun.com/beans/FAQ.html

Client/Server Basics,
http://www.vbhow.to/

Writing COM Objects with Scripting Languages
http://msdn.microsoft.com/scripting/default.htm