

UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS



**Design and Development of
A Bookmark Manager for the University of Nairobi E-learning
System**

Submitted by

**Nancy Njeri Muiruri
P/56/P/7723/02**

**Supervisor:
Dr. E. Omwenga**

Submitted in part fulfillment of the Degree of Master of Science in Information Systems

University of Nairobi.

April, 2009


University of NAIROBI Library



0525354 7

Declaration

This project as presented in this report is my original work and has not been presented for any other university award.

Signed..........

Date.....21st July...2002..

Nancy Njeri Muiruri

Registration Number: P56/P/7723/02

This project has been submitted as part of the fulfillment of the requirements for the Masters Degree in Information Systems of the University of Nairobi with approval as the University supervisor

Signed..........

Date..........

Dedication

To

My parents

For believing in me

To

My God

For helping me believe in myself

Acknowledgement

I would like to thank God for making it possible for me to get an opportunity to pursue a Masters degree course at the university. I would also want to thank my Mum, Dad and family for every provision they made and their continued support throughout this time.

Special thanks to my Supervisor Dr. Elijah Omwenga for his guidance and patience throughout this project.

Lastly, I would like to thank Prof. Anthony Rodrigues for his words of encouragement and insight throughout the Masters course.

Abstract

Bookmarks are normally considered to be thin markers commonly made out of paper or leather, used to mark one's progress in printed material, bookmarks are dated as far back as the sixteenth century.

Automating of bookmarks is dated as far back as with the introduction of the mosaic browser whereby a user would be able to store addresses of pages of interest to him on his personal machine, however unless he imports or exports his bookmarks, they would not be able to be accessed from any other location, hence bringing up the need to have web-based driven bookmark managers.

Live bookmarks, web-based bookmarks or internet bookmarks are database driven bookmark managers, whereby a user may be able to keep a record of important pages they find on the web and would like to resume back to in the future from a browser machine located anywhere.

The University of Nairobi has an E-learning system referred to as Wedusoft, this project is designed to keep a record of all important pages they find on the web as well as to store the last page the user was as the home page, every time he logs out.

DEFINITION OF TERMS

Internet

The internet is a worldwide network of computer networks. It is an interconnection of large and small networks around the globe. The Internet began in 1962 as a resilient computer network for the US military and over time has grown into a global communication tool of more than 12,000 computer networks that share a common addressing scheme

E-learning

E-learning can be defined as any form of learning that utilizes a network for delivery, interaction or facilitation. The network could be the internet, school or college LAN or even a corporate WAN the learning could take place individually (guided by an instructor) or as part of a class. Online classes meet either synchronously (at the same time) or asynchronously (at different times) or as some combination of the two.

Web-based Bookmarks

This is a user's reference to a document on the world-wide web or other hypermedia system; usually in form of a title or comment string and a URL (acronym for "Uniform Resource Locator"). It is the address of a resource on the internet. WWW Addresses begin with <http://>. An example of a URL is <http://www.yahoo.com/>.

In the context of an on-line course, a bookmark is a mechanism that allows a learner to mark their progress through course materials, so that she may interrupt a course and resume studies at a later point.

Web Browser

A web browser is a software application which enables a user to display and interact with text, images, videos, music, games and other information typically located on a

Web page at a website on the World Wide Web or a local area network. Text and images on a Web page can contain hyperlinks to other Web pages at the same or different website. Web browsers allow a user to quickly and easily access information provided on many Web pages at many websites by traversing these links.

HTML

stands for Hyper Text Markup Language; it is the predominant markup language for Web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as links, headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of tags, surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code (such as JavaScript) which can affect the behavior of Web browsers and other HTML processors

JavaScript

It is a scripting language widely used for client-side web development. It is a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but be easier for non-programmers to work with. . JavaScript can interact with HTML source code, enabling Web authors to spice up their sites with dynamic content

TABLE OF CONTENTS

1.0	INTRODUCTION.....	5
1.1	BACKGROUND TO STUDY	6
1.2	PROBLEM DEFINITION.....	7
1.3	OBJECTIVES	7
1.4	PROJECT JUSTIFICATION	7
1.5	PROBLEM SOLUTION	8
1.6	STRUCTURE OF THE DISSERTATION	9
2.0	LITERATURE REVIEW.....	10
2.1	INITIAL DEFINITION OF BOOKMARKS	10
2.2	LIVE BOOKMARKS OVERVIEW (CASE STUDY).....	10
2.3	WEB-BASED BOOKMARKS/LIVE BOOKMARKS	11
2.4	EXAMPLE OF BOOKMARK MANAGERS	12
2.4.1	<i>MybrowserFavorites</i>	12
2.4.2	<i>MyBookmarks</i>	13
2.4.3	<i>Bookmark Tracker</i>	13
2.4.4	<i>Browser-oriented Bookmark Managers</i>	13
2.5	CONCLUSION TO LITERATURE REVIEW	14
3.0	METHODOLOGY.....	15
3.1	FEASIBILITY OF THE PROPOSED SYSTEM.....	16
3.1.1	<i>Operational Feasibility</i>	17
3.1.2	<i>Technical Feasibility</i>	17
3.1.3	<i>Constraints</i>	17
3.2	REQUIREMENT ANALYSIS	17
3.2.1	<i>User interface requirements</i>	17
3.2.2	<i>Processing requirements</i>	18
3.2.3	<i>Output requirements</i>	19
3.3	REQUIREMENTS SPECIFICATION.....	19
3.3.1	<i>Overview of functional requirements</i>	19
3.3.2	<i>Overview of Data requirements</i>	19
3.3.3	<i>General Overview of Requirements</i>	20
3.3.4	<i>General constraints and dependencies</i>	21
3.4	LOGICAL SYSTEM SPECIFICATION.....	21
3.4.1	<i>Data Flow Modeling</i>	22
3.4.2	<i>Entity Behavior Modeling</i>	26
3.4.3	<i>Logical Data Flow Modeling</i>	27
4.0	SYSTEM DESIGN.....	29
4.1	ARCHITECTURE.....	29
4.2	DATABASE DESIGN.....	31
5.0	RESULTS.....	33
5.1	IMPLEMENTATION.....	33
5.2	CHOICE OF PROGRAMMING LANGUAGE.....	33
5.2.1	<i>MySQL</i>	33
5.2.2	<i>PHP</i>	34
5.2.3	<i>HTML and JavaScript</i>	34
5.3	TESTING.....	34
5.3.1	<i>Test Process</i>	35
	<i>Unit Testing</i>	35

<i>System Testing</i>	36
<i>Acceptance Testing</i>	37
5.4 TEST RESULTS	37
6.0 CONCLUSION AND RECOMMENDATIONS	39
7.0 APPENDIX	40
7.1 APPENDIX A: REFERENCES AND BIBLIOGRAPHY	40
7.2 APPENDIX B: USER MANUAL	41
7.3 APPENDIX C: SAMPLE CODE	46

TABLE OF FIGURES

Figure 1 The Classic Waterfall Model.....	16
Figure 2 Bookmark Manager Subsystem.....	20
Figure 3 Context DFD (level 0).....	22
Figure 4 level 1 DFD	23
Figure 5 Log-in DFD (level 1).....	24
Figure 6 Bookmark Manager Subsystem (level 2 DFD)	25
Figure 7 Logical data flow modeling.....	27
Figure 8 Logical data flow modeling (continued)	28
Figure 9 Entity life history (1)	30
Figure 10 Entity life history (2)	31
Figure 11 Testing stages	35

LIST OF TABLES

Table 1: Schema.....	31
Table 2 Subscribers table.....	32
Table 3 User_name table.....	32
Table 4 Url_of_site Table.....	32

1.0 INTRODUCTION

The University of Nairobi has an E-Learning site referred to as Wedusoft, where students can log-in and access course material. E-Learning is any form of learning that utilizes a network for delivery, interaction or facilitation. The network could be the internet, school or college LAN or even a corporate WAN the learning could take place individually (guided by an instructor) or as part of a class.

Wedusoft is an E-Learning platform designed for the University of Nairobi. It consists of a back-end module, which supports server functionalities and houses the representation module and the front end module for learner interaction with the system. E-Learning is a platform designed to empower learners with the skills and knowledge they need in a convenient time and space – independent means. It is the on-line delivery of information, communication, education and training providing a new set of tools that can add value to all traditional learning modes - classroom experience, textbook study, CD-Rom and traditional computer based training. E-Learning does not replace the classroom setting, but enhances it, taking advantage of new content and delivery technologies to enable learning.

The student, who is using the E-Learning site, may want to keep a record of all the pages he may find to be of importance. This activity is referred to as bookmarking. Hence the need to have a system that stores the addresses of these pages (URLs). This study is also designed to come up with a facility that enables the user to store a homepage, preferably the last page that the user was before he logged out.

1.1 Background to study

Students and lecturers on the E-Learning site may be able to access course material on the internet. Students willing to go through the material insert a password for authentication; however a student may not complete the material he/she is going through without interruption. This interruption maybe due to the fact that he has a lecture to attend, due to time constraints or maybe he may get tired and wish to complete the task at a later date. When he resumes back to the course material, he may not remember where he left. This study is attempting to come up with an on-line course bookmark manager that a student may use to store the URLs of the page he last left or those pages he finds to be of importance.

Live bookmarks or web-based bookmarks are pointers - primarily to URLs built in the various internet web browsers. Bookmarks have been incorporated into almost every browser since the mosaic browser and are normally stored on the software client. A folder metaphor may be used for organization. Various shareware utilities and server-side web utilities have been developed to better manage bookmarks, yet none has gained widespread acceptance.

Other definitions of bookmarks are:-

Placeholders for saving web addresses or URLs. The phrase was coined by the designers of Netscape Navigator. Microsoft Internet Explorer uses the somewhat more quaint term "favorites"

A set of stored URLs that can be used by a web client to connect to selected information resources on the internet.

User defined lists that allow a person to easily display a certain location within a document or a page on the WWW.

1.2 Problem Definition

The purpose of this study is to come up with a system that will enable a user to better store his bookmarks as well as edit, delete and place them into folders. It is ideal particularly for researchers who want to keep a record of all the pages they may find to be of importance. The system will also enable the user to store a homepage, the last page he was at before he logged out so that he may be able to return back to it with ease when he logs back into the system.

1.3 Objectives

The objectives of the study include:-

- To enhance the already existing system by bringing up a method of storing URLs (bookmarks), hence allowing a user to organize course material for purposes of research.
- To facilitate a user to return back to the last page he was at before he logged out.
- To provide a system that can be accessed from any location with internet connectivity, rather than one that resides on the client machine and bookmarks have to be imported or exported.

1.4 Project Justification

In the context of an on-line course, a bookmark is a mechanism that allows a learner to mark their progress through course materials, so that she may interrupt a course and resume studies at a later point. This study is designed to research and design a database driven server bookmark manager for the institute.

Bookmarks can be extremely useful in providing quick and easy access to websites you visit regularly or websites where web addresses cannot be easily remembered. The problem

is that you may build up several sets, one at work, another at home, and another at your notebook. This can be frustrating if for example, you find and bookmark a website at home and need to access it at work. It can also be an issue if you have to use another computer, maybe a friend's or when in an internet café.

Live bookmarks will allow you to maintain all your favorite and relevant course content links which can be sorted in groups. This program will make an html page with all your groups of links listed in order, with the date each link was updated, as well as the date the page itself was updated. You can then set this html page as your browser's home page.

Some of the services that existing bookmark managers provide include:-

- Keep your bookmarks secure in your own private password-protected account; therefore one has to register so as to reserve personal database space.
- To keep track of all new, interesting content you discover on the on-line E-learning system.
- To point other web users to your favorite sources of information.
- To organize your course material for purposes of research.
- Keep bookmarks in sync across your different computers and operating systems
- No manual importing and exporting required. E.g. save on disk and transfer it to another browser.
- Supports multiple browsers and operation systems
- One has to register so as to reserve personal database space.
- Include a comprehensive, full-featured bookmark manager facility.
- Effortlessly upgrade your new computer's bookmarks

1.5 Problem solution

A bookmark manager can be able to capture all the requirements that a user may need to organize his work. Bookmark managers are facilities that allow users to mark important

pages or pages whose addresses would not be easily remembered. This study is designed to come up with a bookmark manager for the UON E-learning system (Wedusoft).

1.6 Structure of the dissertation

The following is a description of the areas covered in this dissertation:-

Chapter 2: Literature review that covers different sources of information that discuss bookmark managers. It gives the initial definition of the term 'bookmarks' and how they have been automated.

Chapter 3: The methodology that was used in this project is discussed; this is the Structured Systems and Design Methodology that implements the classic waterfall model.

Chapter 4: The architecture that is used is discussed, outlining the inputs and outputs into the system as well as the structure of the database.

Chapter 5: Covers the results and findings, implementation, choice of programming language and sample of output from the system.

Chapter 6: Summarizes the project and provides a conclusion and recommendation for further work that may be carried out.

2.0 LITERATURE REVIEW

This study is carried out to identify the benefits of having a bookmark manager on the UON E-Learning site. It outlines the initial definition of bookmarks, the concept of automating these bookmarks and concludes by giving the benefits of having a bookmark manager on the UON E-Learning site.

E-Learning has a wide range of advantages, such as reducing time spent commuting to class for students, a reduction on travel costs, one can be able to have a job while taking classes, the student can learn when he needs to (at his convenience), the learning options are not constrained to geographical locations, the attention is focused more on the learner and less on the instructor, lowers cost for both learning providers and organizations that need training (on the long run) and provides an opportunity for both students and lecturers to learn new technologies and skills.

2.1 Initial definition of bookmarks

A bookmark was initially considered to be a thin marker, commonly made from paper or leather, used to keep one's place in a printed work and so be able to return to it with ease some time in the future. Other frequently used materials for bookmarks are leather, metals like silver and brass, silk, wood and fabrics. Many bookmarks can be clipped on a page with the aid of a page-flap..

As the first printed books were quite rare and valuable, it was determined early on that something was needed to mark one's place in a book without causing its pages any harm. Some of the earliest bookmarks were used at the end of the sixteenth century and Queen Elizabeth 1 was one of the first people to own one. See wikipedia

2.2 Live Bookmarks overview (Case Study)

Links are as old as the Web itself. Indeed, links are the stuff of the Web. But the idea of organizing and managing links systematically rather than just listing them on a given web page seems to have emerged with the earliest graphical browsers, and, in particular, with

the foremost browser of its time, Mosaic, the granddaddy of all modern browsers. The simple discipline of presenting link directories arranged on a menu page in nested list form harks back to earlier, more rigid information systems such as Gopher. The Web, however, was supposed to be different with nodes of information connected in a truly open, free-form manner rather than being accessible only by navigating a strict, pre-determined path hierarchy within a single authority domain. Mosaic developed a feature called Hotlists, which, while still hierarchical and aping the common file system paradigm of folders and files, at least allowed for links to be easily recorded and for ready access to any recorded link from any page within the browser. By the time the initial development crew had decamped from NCSA to set up a commercial operation and built the new, line-in-the-sand Netscape browser, this feature had become reincarnated as Bookmarks.

Meanwhile, Microsoft, finally catching the wave, was engineering a vigorous response to the new, upstart Netscape browser. This was to be called Internet Explorer and included a similar link manager that was dubbed Favorites. Bookmarks or Favorites had now become an integral part of users' everyday web experience and would remain so until powerful search engines such as Google and Yahoo! came knocking on the door. Because, while bookmarks allowed users to record sites of interest, they soon grew to become unwieldy in terms of needing to be managed within the confines of a simple, hierarchical structure. It became apparent with the growing power of the new search engines that it was easier just to search for a particular site afresh each time – in effect, search engines were now able to provide a dynamic book marking service, or bookmarks on demand. See Social Bookmarking Tools, Nature Publishing Group

2.3 Web-based bookmarks/Live Bookmarks

This refers to computerized or automated bookmarks, where a user places addresses of those pages he finds to be importance on a computer. These addresses can then be accessed from any location as long as the machine has internet connection.

Bookmarks are navigational tools that facilitate the reader's perusal of a document. The bookmarks can be hierarchically structured (nested and linked to pages in the document.) The reader need only to click on a bookmark to be sent to the page where the bookmark is linked.

Bookmarks have been incorporated in browsers since the Mosaic browser in 1993. Bookmark lists were called Hotlists in Mosaic and in previous versions of Opera; this term has faded from common use. Other early web browsers such as ViolaWWW and Cello also had bookmarking features. See wikipedia

Nesting of bookmarks provides an efficient way to display a comprehensive list of bookmarks in a small amount of space without sacrificing document navigability. Bookmarks can be organized and placed in folders, thus accessed collectively within their respective folders.

2.4 Example of Bookmark Managers

The following are examples of bookmark managers:-

2.4.1 MybrowserFavorites

The most relevant information stored on the World Wide Web where an individual user is concerned can be bookmarked (URLs of pages are stored on a folder).

This sidebar is accessible every time the user logs on and wishes to browse the internet.

With MyBrowserFavorites you can manage favorites online, allowing them to be accessed wherever you have internet connection, whether that's at work, at home, on a notebook, or in an internet café. See www.mybrowserfavorites.com

2.4.2 MyBookmarks

This is an example of a free internet service that allows one to keep his browser bookmarks and favorites on-line so that they can be accessed from anywhere. Internet Explorer favorites can be imported to MyBookmarks to get started quickly.

A full-featured editor makes it easy to organize and search for bookmarks on-line. You can even export your on-line bookmarks back to your browser. See www.mybookmarks.com

2.4.3 Bookmark Tracker

This is another example of a bookmark manager that allows one to safely store and organize bookmarks on-line and securely access them from anywhere on the internet.

Major Features/Benefits include:

- Keep your bookmarks secure in your own private password-protected account.
- No banner ads, popup ads or email spam messages.
- Synchronize your bookmarks securely between multiple computers and/or browsers.
- Share your bookmarks anonymously with others.
- Completely web-based. No additional software to install.
- Store an unlimited number of bookmarks.
- Automatic folders (My New Bookmarks, My Top Rated Bookmarks, Recently Visited, Most Visited)
- Fast and FREE!
- Works with Netscape, Mozilla, Firefox and Internet Explorer

See www.bookmarktracker.com, 1998-2006

2.4.4 Browser-oriented Bookmark Managers

With Mozilla Firefox bookmarks are simply referred to as bookmarks on the toolbar.

When accessing bookmarks with Mozilla Thunderbird, we simply go to tools on the toolbar then account.

Favorites is used with Internet Explorer where one is able to create a personal folder, manage it as well as add and edit bookmarks. This is the commonly used term for the bookmark manager on Microsoft Internet Explorer.

2.5 Conclusion to literature review

A bookmark Manager is a facility designed to enable a web user to keep track of the URLs for all interesting pages they may find on the web. Live bookmarks sit in the bookmarks menu or sidebar. Bookmark managers allow users to add or manage his bookmarks, they may be browser oriented or those managed by different companies or learning institutions.

These bookmark managers provide user defined lists that allow a person to easily display a certain location within a document or a page, they are ideal where a user is carrying out research work on the internet or when he wants to store up URLs for pages that he would like to resume to after logging out. This study is to facilitate the same activities for any person who has logged onto the University of Nairobi E-learning system. He will be able to organize his course material according to his needs.

3.0 METHODOLOGY

This project was structured in distinct phases with defined deliverables from each phase. The methodology used is the Structured Systems Analysis and Design Methodology (SSADM) which implements the classic waterfall model.

The overall idea of using this methodology was so that all system development activities are done in a coordinated manner as well as allow for monitoring of time, resources and budget. In addition to this we expected better report documentation because every stage was documented.

Primarily, a waterfall methodology structures a project into distinct phases with defined deliverables from each phase. The phases are always called something different depending on which company flavor, but the basic idea is the same.

SSADM application development projects are divided into five modules that are further broken down into a hierarchy of stages, steps and tasks:-

Feasibility Study where the current system is analyzed to determine whether it can support the additional requirement.

Requirements Analysis where the requirements of the system to be developed are identified and the current system is modeled in terms of the processes carried out and the data structures involved.

Requirements Specification where detailed functional and non-functional requirements are identified and new techniques are introduced to define the required processing and data structures.

Logical System Specification where technical systems options are produced and the logical design of update and enquiry processing and system dialogues.

Physical Design where a physical database design and a set of program specifications are created using the logical system specification and technical system specification.

The classic Waterfall Model

The figure below illustrates the Classic Waterfall Model which is the general conceptual framework used in this study, laying emphasis on the SSADM stages

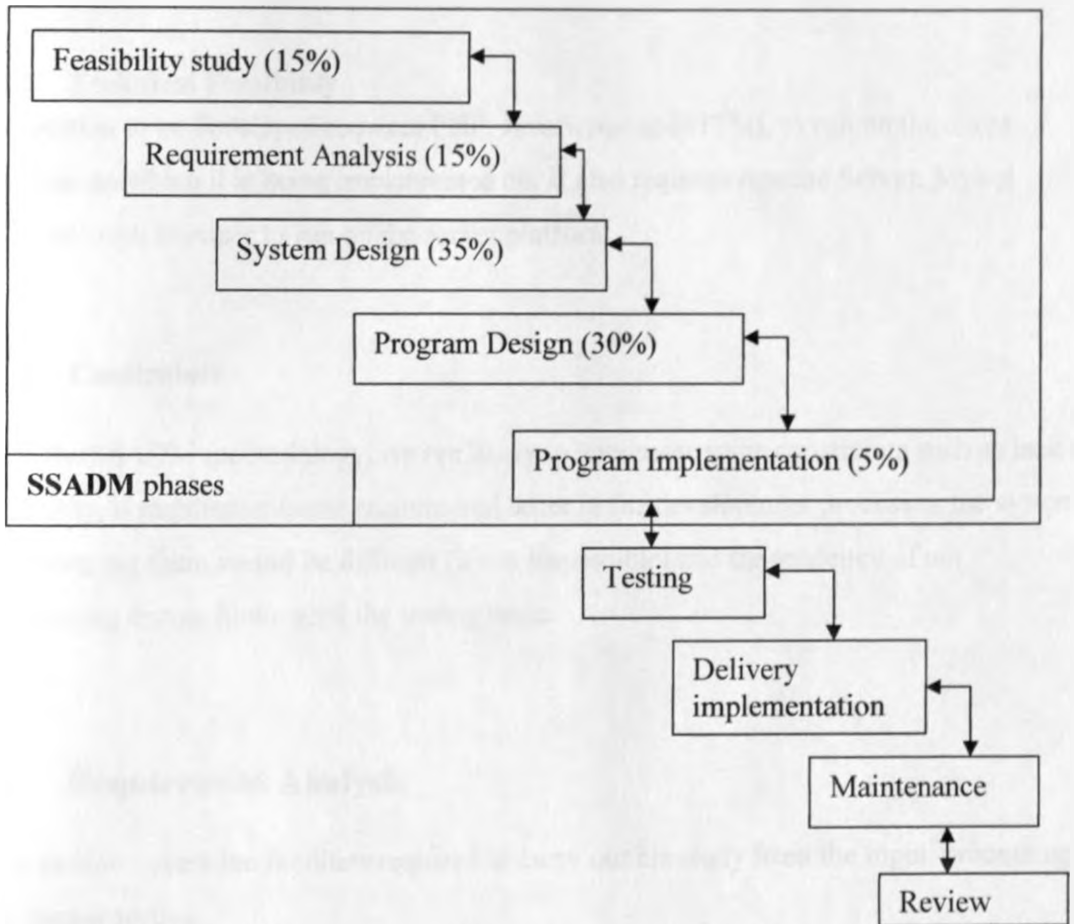


Figure 1The Classic Waterfall Model

3.1 Feasibility of the proposed system

This is the measure of how beneficial or practical the development of an information system will be to an organization, in this case feasibility has been assessed in terms of its practicality, technical and time factors.

3.1.1 Operational Feasibility

An overview with a Subject Matter Expert (SME) of the University of Nairobi E-learning system portrayed that incorporating a bookmark manager on the current functional system would be an added advantage. It would facilitate a student's perusal through documents, particularly researchers to keep their bookmarks in an organized manner. On logging in the user would be referred to the last page he was at before he logged out.

3.1.2 Technical Feasibility

The system to be developed requires PHP, JavaScript and HTML to run on the client machine on which it is being implemented on. It also requires Apache Server, MySQL database, web browser to run on the server platform.

3.1.3 Constraints

With the SSADM methodology, we are likely to encounter some constraints such as lack of flexibility, if requirements are encountered latter in the development process or the system incorporating them would be difficult (if not impossible) and the tendency of not discovering design flows until the testing stage.

3.2 Requirement Analysis

This section covers the facilities required to carry out his study from the input, processing and output section.

3.2.1 User interface requirements

a) Web Browser

The web Browser that was used in this project was Microsoft Internet explorer, although the system was also tested on Mozilla Firefox.

b) HTML

It is the predominant markup language for Web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as links, headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects.

c) JavaScript

It is a dynamic, weakly typed, prototype-based language with first-class functions. It is used in this case to validate whether a user has entered a given field and does so by use of alerts.

3.2.2 Processing requirements

a) Apache Server

Apache, otherwise known as Apache HTTP Server, is an established standard in the online distribution of website services, which gave the initial boost for the expansion of the World Wide Web. It is an open-source web server platform, which guarantees the online availability of the majority of the websites active today. The server is aimed at serving a great deal of widely popular modern web platforms/operating systems such as Unix, Windows, Linux, Mac OS X, Microsoft Windows, OS/2, etc.

b) MySQL database.

MySQL is a relational database management system (RDBMS) which has more than 11 million installations. The program runs as a server providing multi-user access to a number of databases. Its popularity for use with web applications is closely tied to the popularity of PHP, which are often combined with MySQL.

c) PHP

PHP is a scripting language originally designed for producing dynamic web pages. It has evolved to include a command line interface capability and can be used in standalone graphical applications. PHP generally runs on a web server, taking PHP

code as its input and creating web pages as output. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

d) HTML and JavaScript.

3.2.3 Output requirements

The web browsers that may be used to run this system are Microsoft Internet Explorer or Mozilla Firefox.

3.3 Requirements Specification

This is a description of the input and output requirements for the bookmark manager system in terms of Functionality and data requirements as well as general constraints and dependencies.

3.3.1 Overview of functional requirements

There are two basic functional requirements:-

- a) Storing the last page as the home page every time the user logs out.
- b) Store bookmarks- store URLs of pages that the user finds to be of importance as well as manage them.

3.3.2 Overview of Data requirements

This is the description of input and output from the product

Input Requirements include:-

- User Log-in details
- URLs for important pages
- Setting homepage

Output Requirements include :-

- An sorted list of URLs
- A home page that is the last page the user was before he logged out.

The data requirements from the user's point of view are requirements that will be required of the user, the user will be asked whether he would like to:-

- Add a URL
- Edit a URL
- Delete a URL
- Display a bookmark
- Create a folder
- Import or export bookmarks from favorites

3.3.3 General Overview of Requirements

The figure below summarizes the input, output and processing requirements carried out in this dissertation.

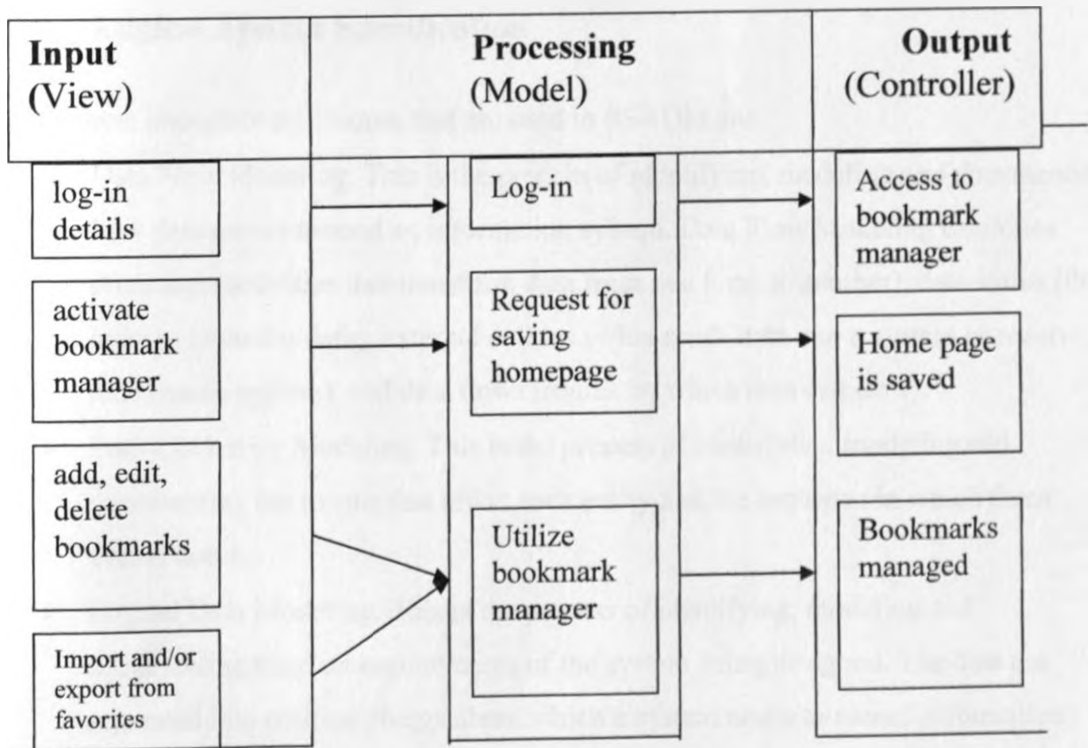


Figure 2 Bookmark Manager Subsystem

3.3.4 General constraints and dependencies

Dependencies

This product is designed to run on the Apache server, it requires a windows environment and a MySql database.

Constraints

The Methodology used in this project is the SSADM methodology, we are likely to encounter some constraints such as:-

- Lack of flexibility; if design requirements are encountered later in the development process they may be difficult (if not impossible) to incorporate.

- There is a tendency of not discovering design flaws until late in the testing stage.

3.4 Logical System Specification

The 3 most important techniques that are used in SSADM are:

- **Data Flow Modeling.** This is the process of identifying, modeling and documenting how data moves around an information system. Data Flow Modeling examines processes (activities that transform data from one form to another), data stores (the holding areas for data), external entities (what sends data into a system or receives data from a system), and data flows (routes by which data can flow).
- **Entity Behavior Modeling.** This is the process of identifying, modeling and documenting the events that affect each entity and the sequence in which these events occur.
- **Logical Data Modeling.** This is the process of identifying, modeling and documenting the data requirements of the system being designed. The data are separated into entities (things about which a system needs to record information) and relationships (the associations between the entities).

This involves decomposition of the whole system into parts in an attempt to study how those parts interact.

3.4.1 Data Flow Modeling.

Step 1

Level 0

The Context DFD below describes the system's relationship with the rest of the world.

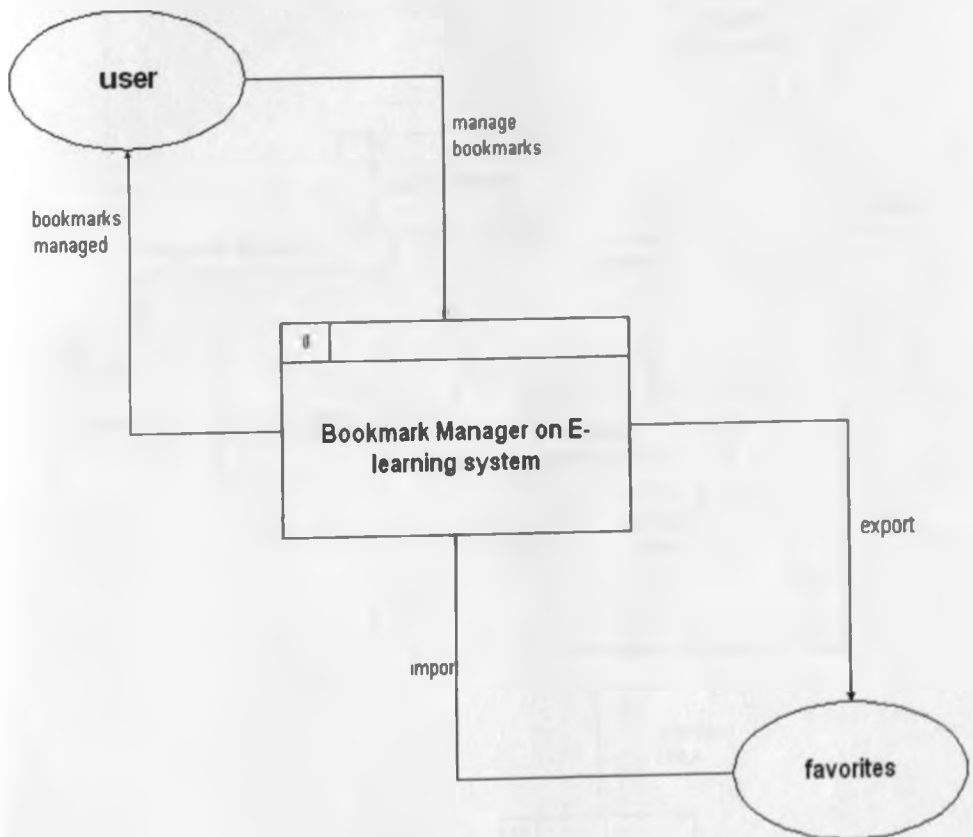


Figure 3 Context DFD (level 0)

Step 2: Bookmark Manager Exploded

Level 1

The figure below outlines an exploded view of the bookmark manager

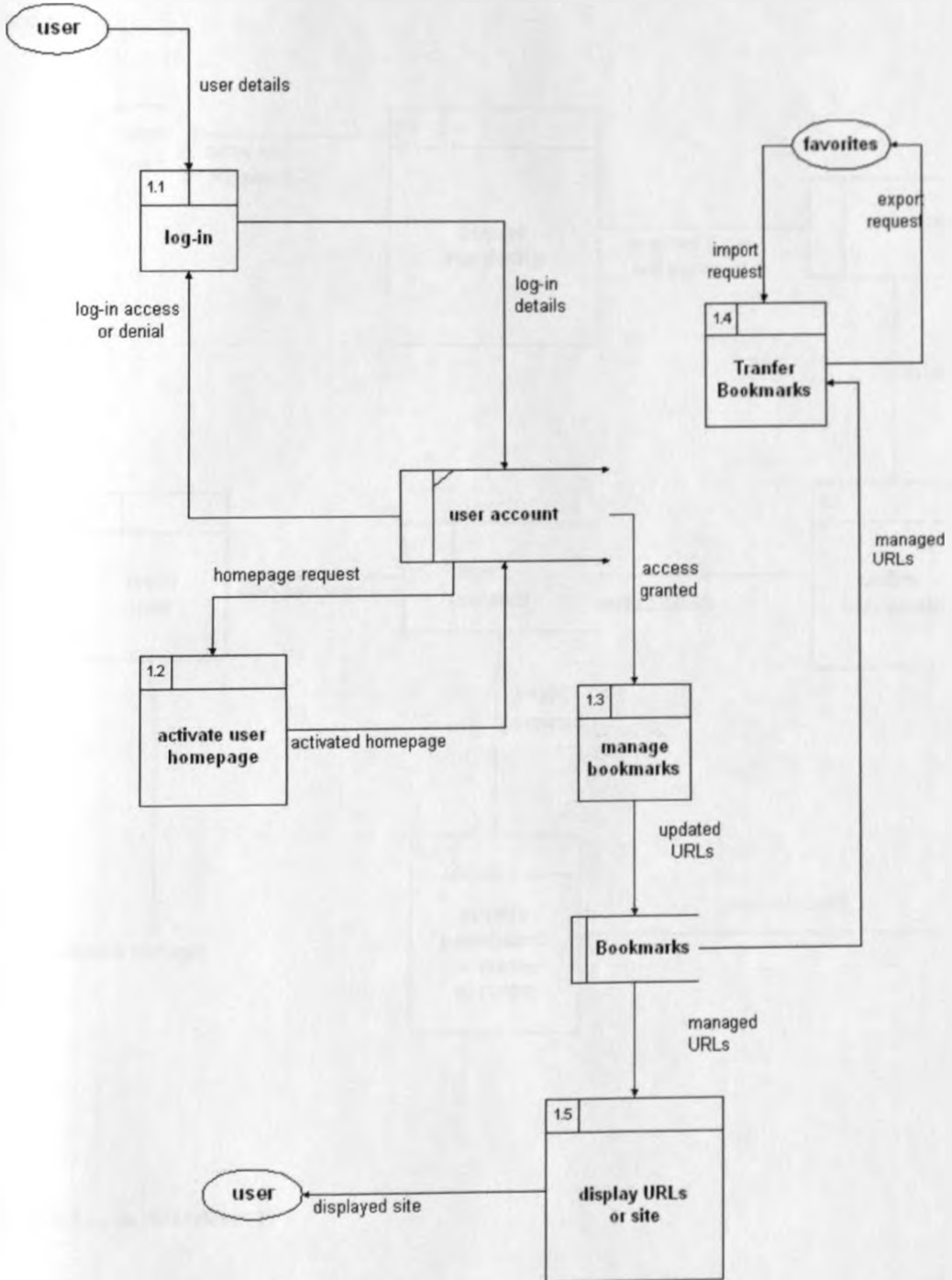


Figure 4 level 1 DFD

Step 3

Level 1 DFD

The figure below outlines the log-in process where the user is either granted or denied access

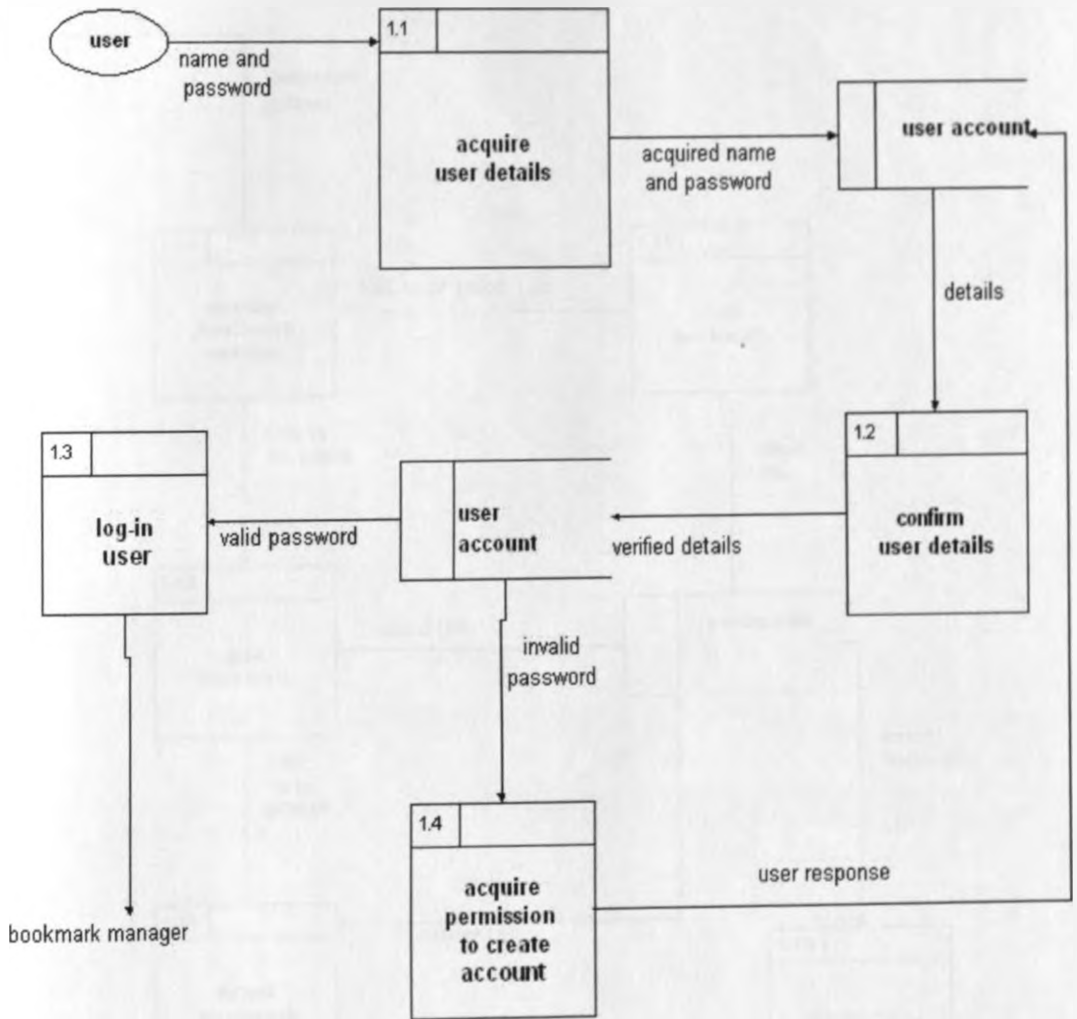


Figure 5 Log-in DFD (level 1)

Step 4

Level 2 DFD

Book Manager Subsystem Exploded

The figure below outlines the actual activities carried out in the exploded version of the system

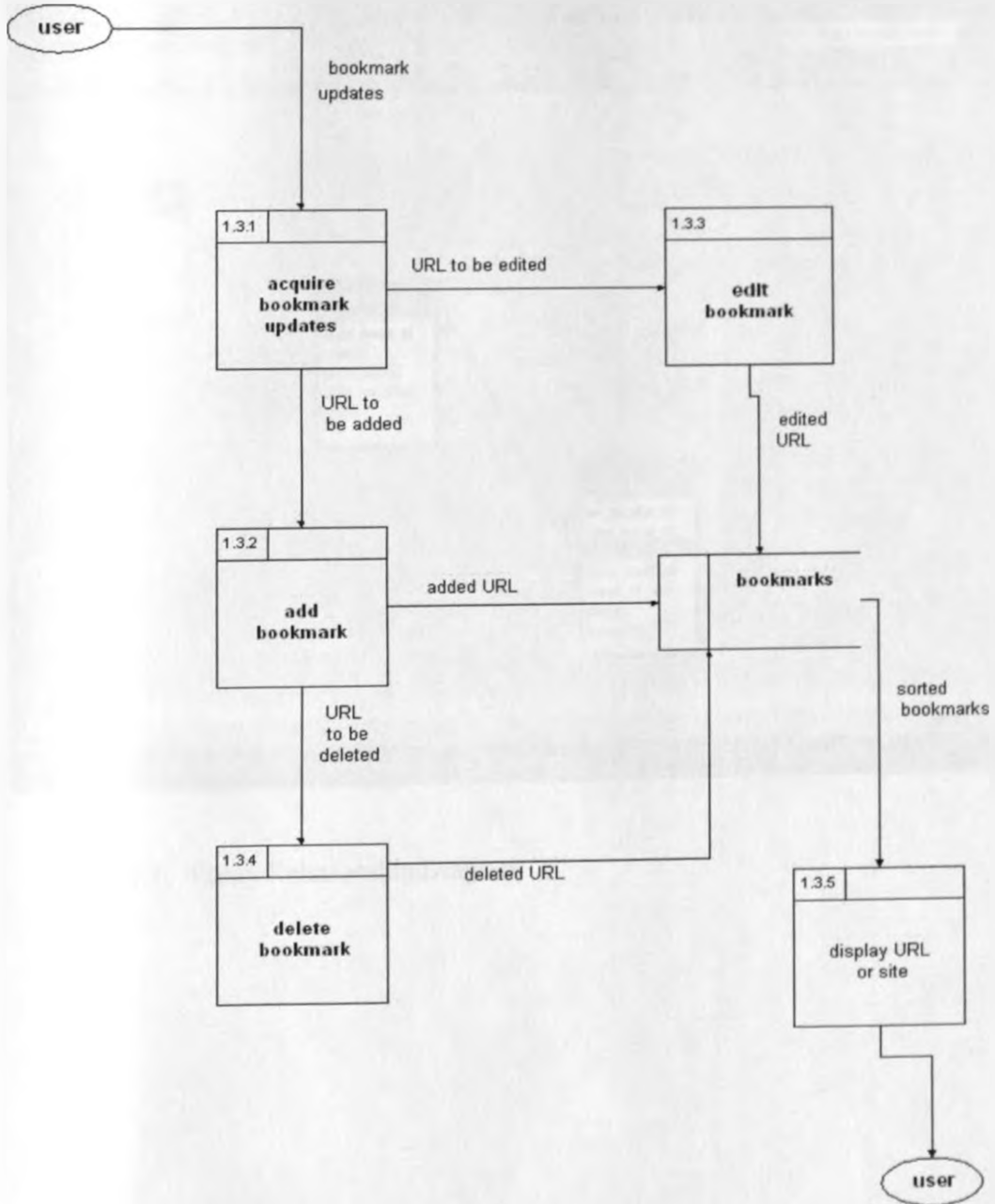


Figure 6 Bookmark Manager Subsystem (level 2 DFD)

3.4.2 Entity Behavior Modeling

Entity Relationship Diagram (ERD) for Bookmark Manager System

Below is the ERD for the system, displaying the entities (tables) and their relationships

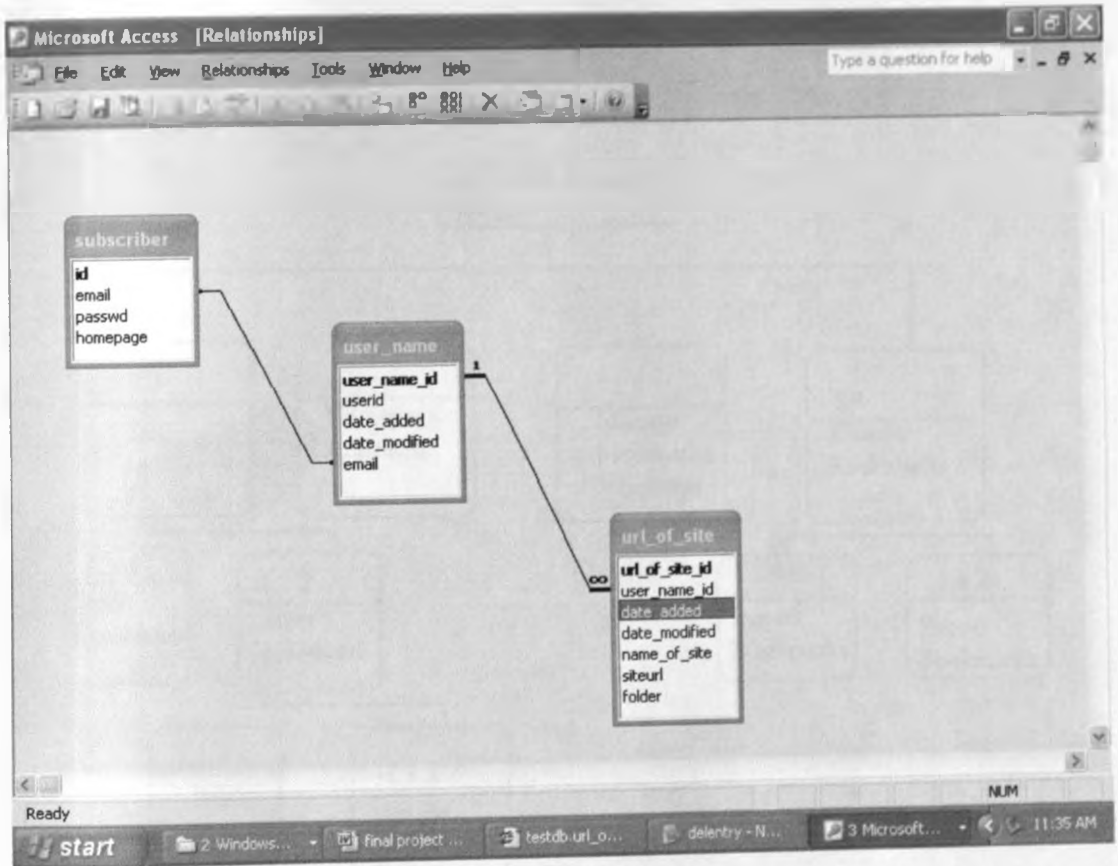


Figure 1. Entity Relationship Diagram

3.4.3 Logical Data Flow Modeling

The figure below illustrates the bookmark manager residing on the UON E-Learning site, a user has to log in so as to reserve database space. Once a user has logged in, he may wish to save his home page, which will automatically be generated the next time he logs in. Then he will utilize the facilities offered by the bookmark manager. Finally he is given an option as to whether or not he would like to import or/and export his bookmarks.

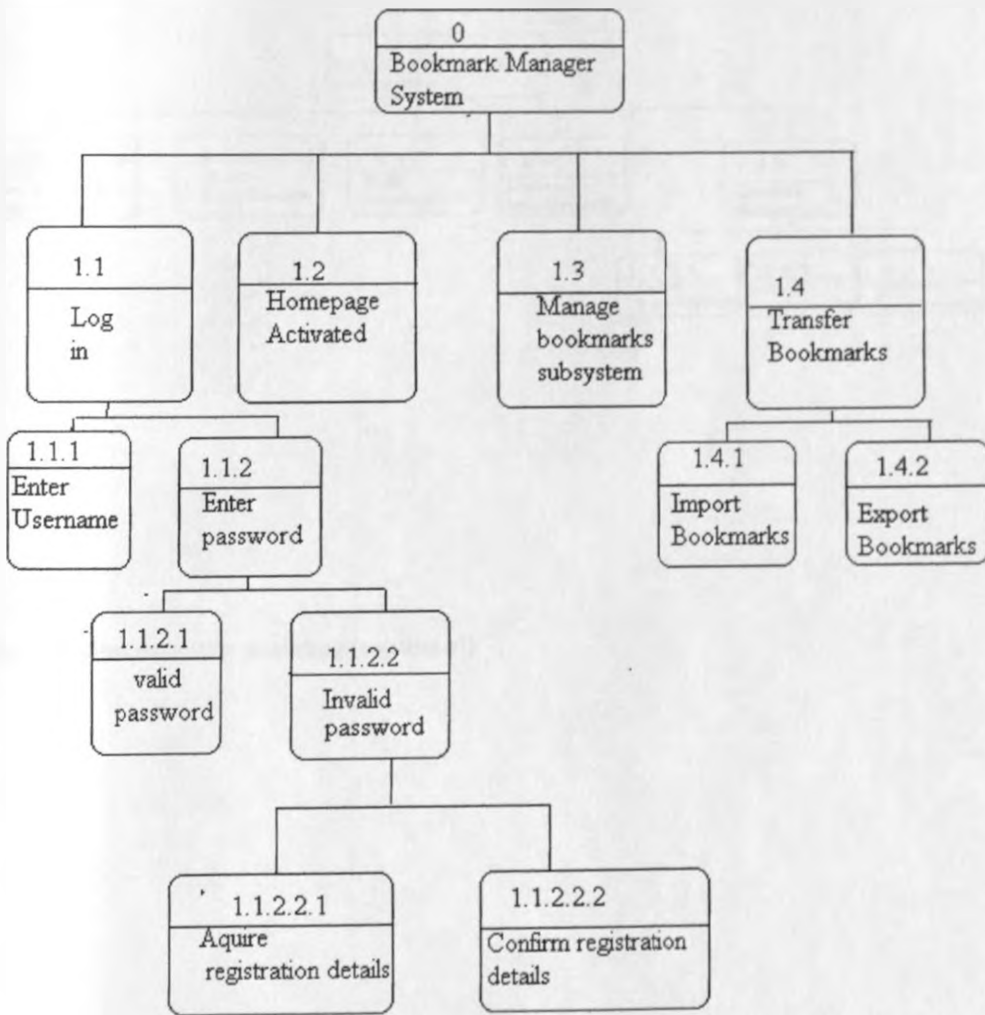


Figure 7 Logical data flow modeling

Logical Data Modeling (continued)

The figure below is an exploded version of the bookmark manager subsystem (item 1.3 in figure 8). Whereby, upon acquiring bookmark updates, he may choose to add, edit or delete his bookmark. He may also have the option of placing of placing his bookmarks in a folder, so that they may be sorted out from there.

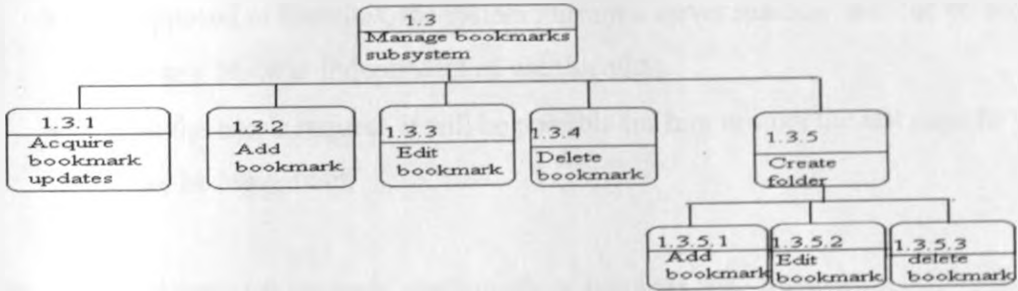


Figure 8 Logical data flow modeling (continued)

4.0 SYSTEM DESIGN

The product that is being designed is a database driven server bookmark manager. It requires running on the Apache server, Windows NT environment and a MySQL database.

It has advantages over the older system in that:-

- The older system does not have a bookmark manager so it will enable the user to store the URLs of those pages he finds to be of importance.
- As opposed to favorites, the system runs on a server machine and can be accessed from any browser independent of user location.
- Upon the user's request, it will be possible for him to store the last page he was before he logged out.

The potential users are students, particularly researchers who are seeking information from the E-Learning system and Lecturers.

4.1 Architecture

This is a client-server system; it resides on the server and can be accessed by a client from any location.

The system will have the following major inputs:-

- User email address
- URLs for important pages
- Saving of homepage

Major outputs include:-

- A bookmarked list of URLs
- A homepage that is the last page the user was before he logged out.

Entity Life History (1)

The figure below outlines all the activities that take place in this system from the design view

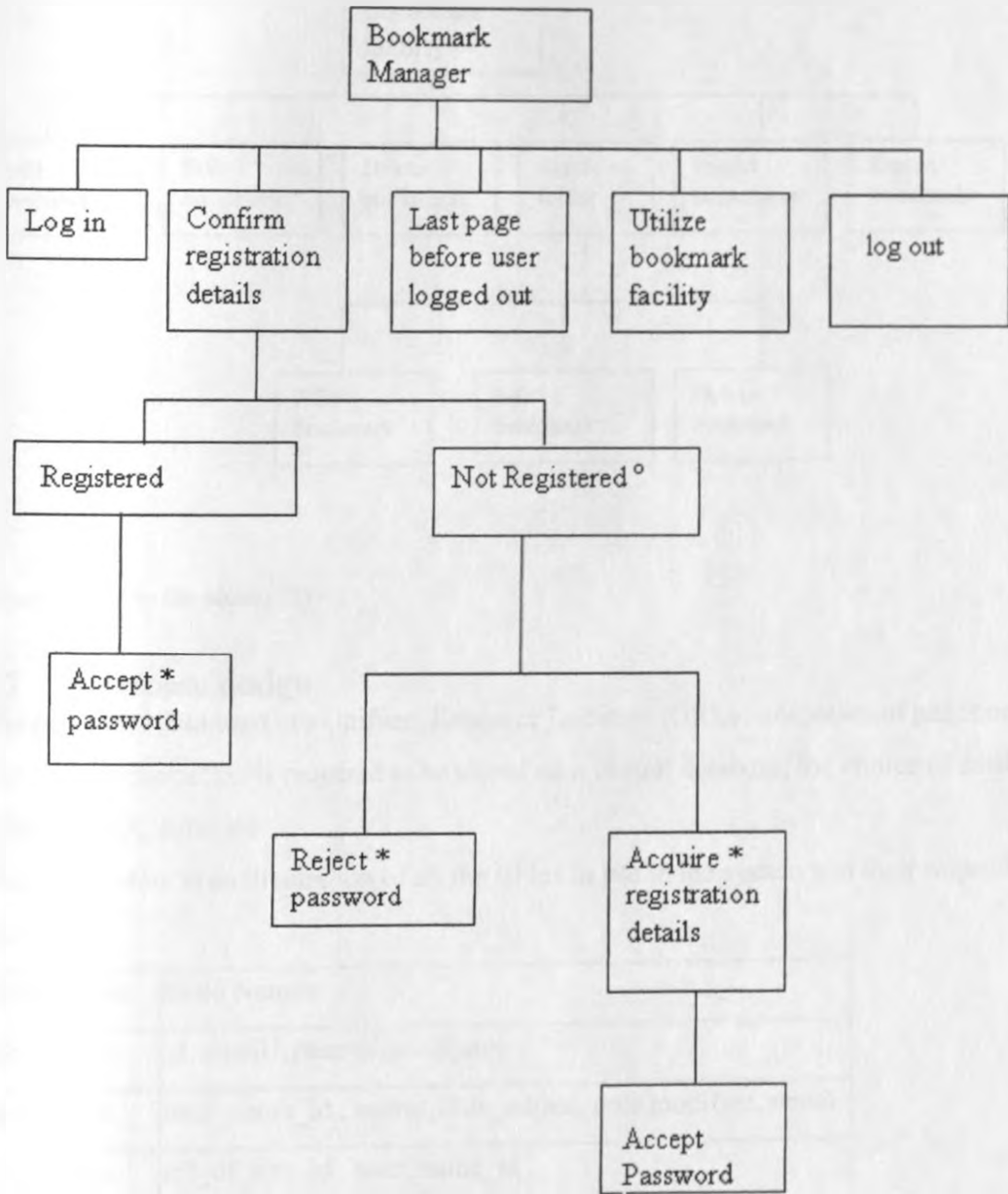


Figure 9 Entity life history (1)

Entity Life History (2)

The figure below outlines an exploded version of all the activities carried out by the bookmark manager

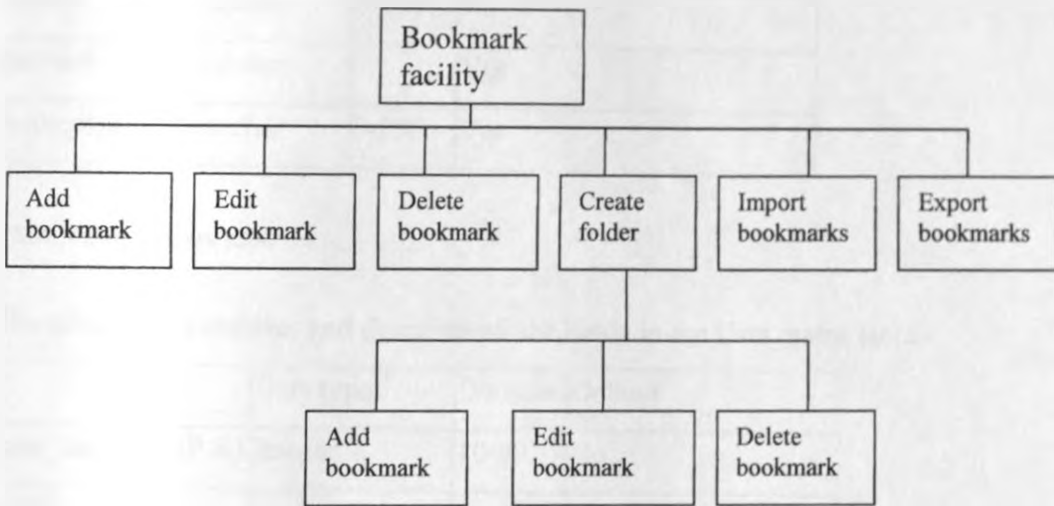


Figure 10 Entity life history (2)

4.2 Database design

The persistent data used are Uniform Resource Locators (URLs)-addresses of pages on the web. This information is required to be stored on a central database; the choice of database is the MySQL database.

The table below is an illustration of all the tables in use in the system and their respective fields

Table Name	Field Names
subscribers	id, email1,passwd,homepage
user_name	user_name_id , userid, date_added, date modified, email
url_of_site	url_of_site_id , user_name_id , date_added,date_modified, siteurl, name_of_site, folder

Table 1: Schema

The table below outlines and describes all the fields in the subscribers table

	Data type	Domain	Default
id	int	0-11	No
email	varchar	0-150	No
passwd	varchar		Yes
homepage	varchar	0-255	Yes

Table 2 Subscribers table

The table below outlines and describes all the fields in the User_name table

	Data type	Domain	Default
user_name_id (P.K)	integer	10-99	No
userid	mediumint	5	No
date_added	datetime		Yes
date_modified	datetime		Yes
email	varchar	0-150	NOT NULL

Table 3 User_name table

The table below outlines and describes all the fields in the Url_of_site table

	Data type	Domain	Default
url_of_site_id (P.K)	integer	0-11	No
user_name_id (S.K)	varchar	0-150	No
date_added	datetime		Yes
date_modified	datetime		Yes
name_of_site	varchar	0-150	No
siteurl	varchar	0-150	Yes
folder	varchar	0-150	Yes

Table 4 Url_of_site Table

5.0 RESULTS

5.1 Implementation

A bookmark Manager Subsystem was built for the currently running E-learning system. A normal user will access the client side of the system

The objectives of this part are:-

- a) Ensuring everything is in place i.e the Software, Hardware and Personnel
- b) System installation
- c) Training of system users
- d) Conducting evaluation of fulfillment of system requirements

The software that was used include:-

- Operating system: windows system, Microsoft Windows XP
- MySQL DBMS
- Microsoft office 2003
- Microsoft Visio 2003
- Microsoft Access 2003

5.2 Choice of programming language

Below is a description of the programming languages used to implement the system:-

5.2.1 MySQL

MySQL is the world's most popular open source database, recognized for its speed and reliability. MySQL AB, the company founded by the creators of the MySQL database, provides MySQL software development and related support and services.

Advantages of using MySQL

- It is one of the best and most used database in the world

- Available and affordable for all
- Easy to use
- Continuously improved while remaining fast and safe
- Fun to use and improve
- Free from bugs

Therefore MySQL was used to create the database.

5.2.2 PHP

PHP stands for 'Hypertext Preprocessor' it is the programming language that is currently in use in the design of the UON E-Learning system, Wedusoft.

It is one of the most popular server side scripting languages running today. It is used for creating dynamic web pages that interact with the user offering customized information.

PHP has the following advantages:

It is fast and easy to learn, especially for those with backgrounds in programming such as C, JavaScript and HTML. PHP is an open source software (free).

5.2.3 HTML and JavaScript

HTML and JavaScript which are client-side scripting languages were also used. HTML is the primary scripting language commonly used to create web pages while JavaScript is used for validation scripts of forms and requesting user inputs.

5.3 Testing

Testing is done to establish the presence of system defects in programs and also to judge whether the program is usable in practice. Testing can also establish the presence of errors. Except for small programs, systems should not be tested as a single unit. Large systems are

built out of modules, which are composed of procedures and functions. The testing is therefore carried out incrementally in conjunction with system implementation.

5.3.1 Test Process

The testing process used consists of the five stages in the figure below:-

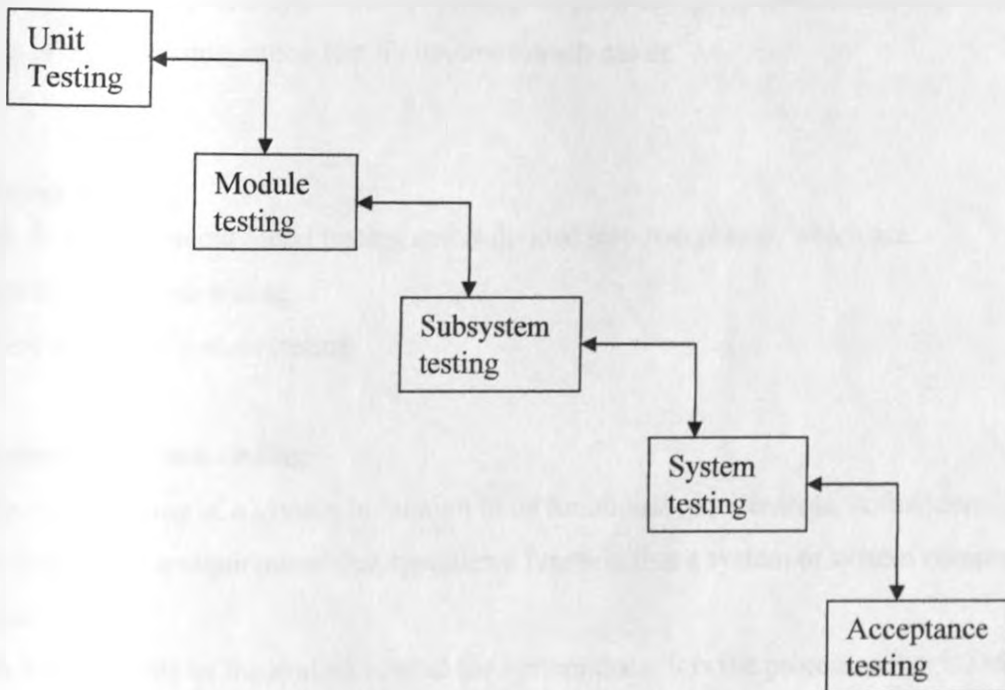


Figure 11 Testing stages

Unit Testing

Unit testing tests the minimal software component, or module. Each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly (i.e. regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.

Readily-available unit tests make it easy for the programmer to check whether a piece of code is still working properly. Good unit test design produces test cases that cover all paths through the unit with attention paid to loop conditions.

Unit testing helps to eliminate uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

System Testing

This is a requirement based testing and is divided into two phases, which are: -

Functional system testing

Non-Functional system testing

Functional system testing

This is the testing of a system in relation to its functional requirements. A functional requirement is a requirement that specifies a function that a system or system component must perform.

The functionality of the system is what the system does. It is the processes that it carries out. Therefore we can say that functional testing is testing to see that the system does what it is supposed to do.

- Tests are derived from user requirements or system requirements.
- Tests are based on documented requirements and not the delivered code.
- Test preparation can occur as soon as the user requirements have been agreed.
- Promotes quality within the finished system.

Business process functional testing is a testing based on expected user profiles. It should reflect the business environment and the business processes.

However, it must be ensured that all functionality is tested within the system.

Non-Functional system testing

This is the testing of everything that doesn't relate to the functionality of the system. This type of testing covers aspects such as ease of use and performance. The system may provide all the necessary functionality but if it is not easy to use or does not perform very well then it will not be "fit for purpose".

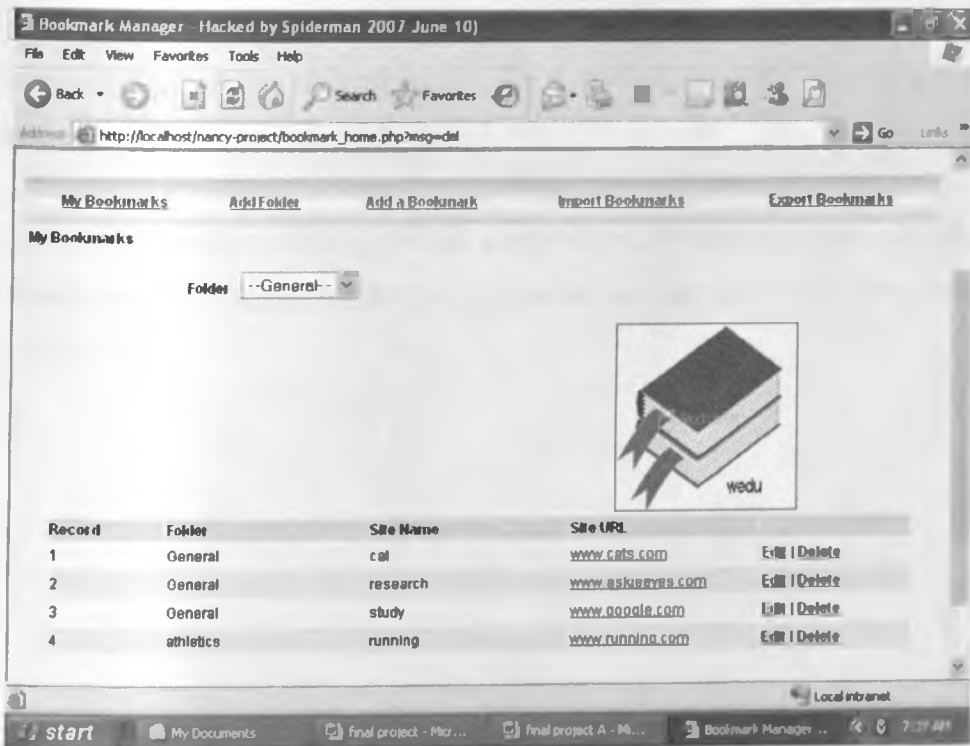
Acceptance Testing

The formal testing conducted to enable a user, customer, or other authorised entity to determine whether to accept a system or component.

Acceptance testing is the testing of deliverable, for acceptability, by the user of that deliverable.

5.4 Test results

Below is a sample of the main page, the page displaying all the user inputs to the system



Record	Folder	Site Name	Site URL	
1	General	cal	www.cats.com	Edit Delete
2	General	research	www.eskervees.com	Edit Delete
3	General	study	www.google.com	Edit Delete
4	athletics	running	www.running.com	Edit Delete

Verification and Validation

Verification and Validation starts with requirements reviews and continues through design and code reviews to product testing. (V&V) seeks to answer the following questions:-

Validation: are we building the right product?

The product that was designed was the database driven server bookmark manager subsystem, which is running on the UON E-Learning server. Validation is the degree to which goals are achieved; it is concerned with the output of the system. In this study, the main outputs are the URL's for specified pages. The system records and displays them appropriately, updating may take place upon user request. The other major output is the display of the user homepage; a separate window is opened every time the user logs in, if he had requested his homepage to be saved.

Verification: are we building the product right?

All the components that were needed to be designed were put in place, adding functionality to the entire system. Verification is the degree to which the inputs are implemented to achieve outputs. In this study, all inputs (URL's) the user specified are systematically output as required. Any updates such as delete, edit or add on the client side is transferred properly to the server side and represented back to the user.

Verification involves checking that the product conforms to its specification whereas validation involves checking that the program as implemented meets the expectations of the customer.

6.0 CONCLUSION AND RECOMMENDATIONS

A Bookmark manager was successfully designed for the currently running University of Nairobi E-Learning system Wedusoft. It will facilitate the users to store important pages as well as save their homepage for later use. It captures all the advantages of having a Bookmark Manger, and in this case it is customized to a user of the E-Learning system.

The bookmark manager has captured and satisfied all the objectives that were initially specified in the following way:-

1. This system has successfully implemented a way of storing user specified URL's for the currently running system, wedusoft. This is particularly ideal for researchers who may want to keep a record of sites that they may find to be of use some time in the future.
2. The web-based bookmark manager can be accessed from any machine with internet connectivity, thus a user does not necessarily have to import and/or export his bookmarks. The system also provides a means for the user to store the bookmarks on his personal machine.
- 3 Storing the last page as the homepage when the user logs out captures the actual concept of the book marking activity, where when you leave a site, you are taken back to the actual page you were on directly when you log back in on a future visit.

Recommendations for further work would be that the database design be incorporated on the already currently running database. This will facilitate uniformity in the user's access to the E-Learning site and all the activities that he will carry on the site. The last page that is normally stored in the database per user session is generated when the user logs in. Further study may take pace such that the last page is stored automatically when the user logs out.

7.0 APPENDIX

7.1 APPENDIX A: REFERENCES AND BIBLIOGRAPHY

Julie, M, "SAMS Teach yourself PHP, MySQL and Apache" (2nd edition) London: Pearson, (2006).

Tim C., Joyce P. and Clark M. "PHP5 and MYSQL Bible" Indiana: Wiley, (2004)

Whitten, Dittman and Barlow, "System Analysis and Design methods" (2nd Edition) New Delhi: Whitten, (1996)

Dittman, Whitten and Barlow. "Systems Analysis and Design Methods" (5th Edition) New York: Mc-Graw-Hill, (2001)

Wikipedia, the free encyclopedia

<http://en.wikipedia.org/wiki/Bookmark>

Jim (2001)mybookmarks-[online] <http://www.mybookmarks.com>

Brian Bray (1998-2006) bookmarktracker-[online] <http://www.bookmarktracker.com>,

Winston (1999) mybrowserfavorites-[online] <http://www.mybrowserfavorites.com>

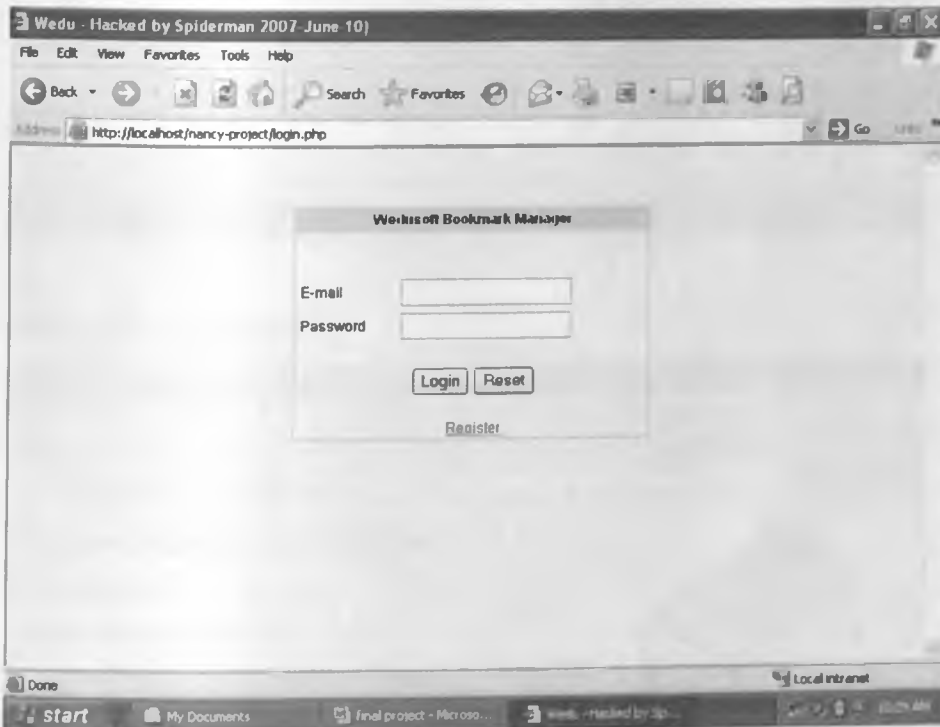
Hammond T, Hannay T.,Lund B. "Social Bookmarking Tools, Nature Publishing Group"- [online] <http://www.dlib.org/dlib/april05/hammond/04hammond.html>

7.2 APPENDIX B: USER MANUAL

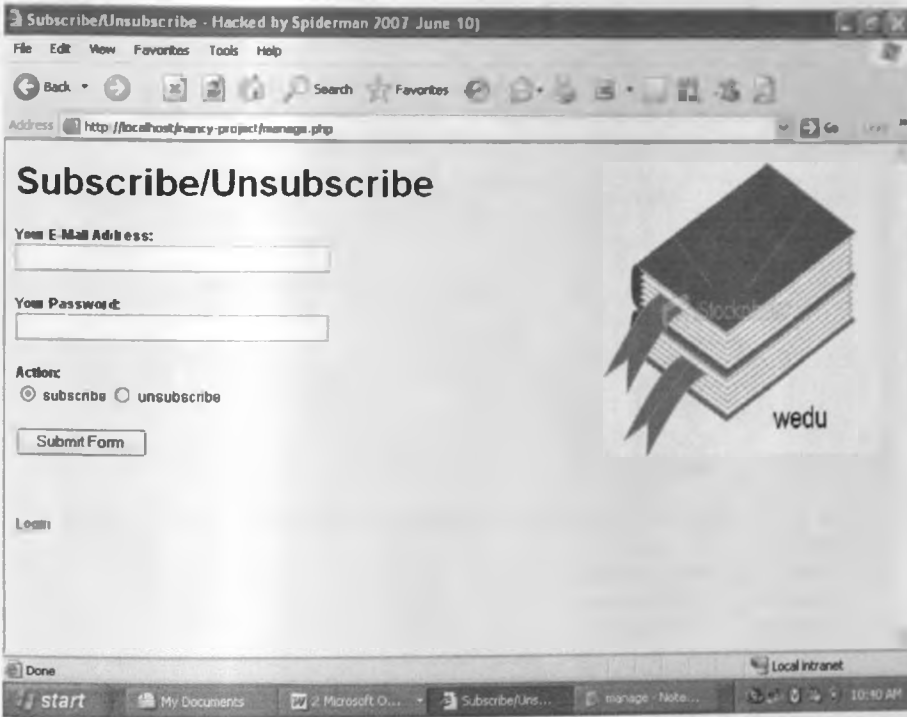
Sample Screens used in the Bookmark Manager System

One has to subscribe so as to use the bookmark manager. The user is expected to input an email address so as it can act as a unique identifier when he is using this facility and so as to enable communication to him incase there are any problems in the system. The e-mail address should also be input so as to enable customized saving of a user's homepage, upon request.

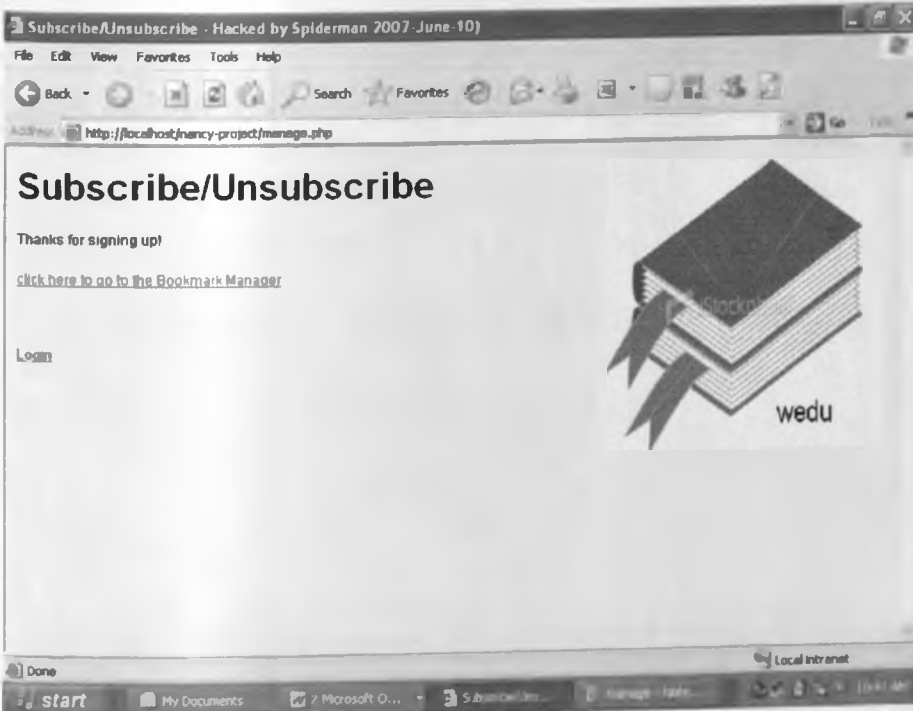
Initial Screen: login.php



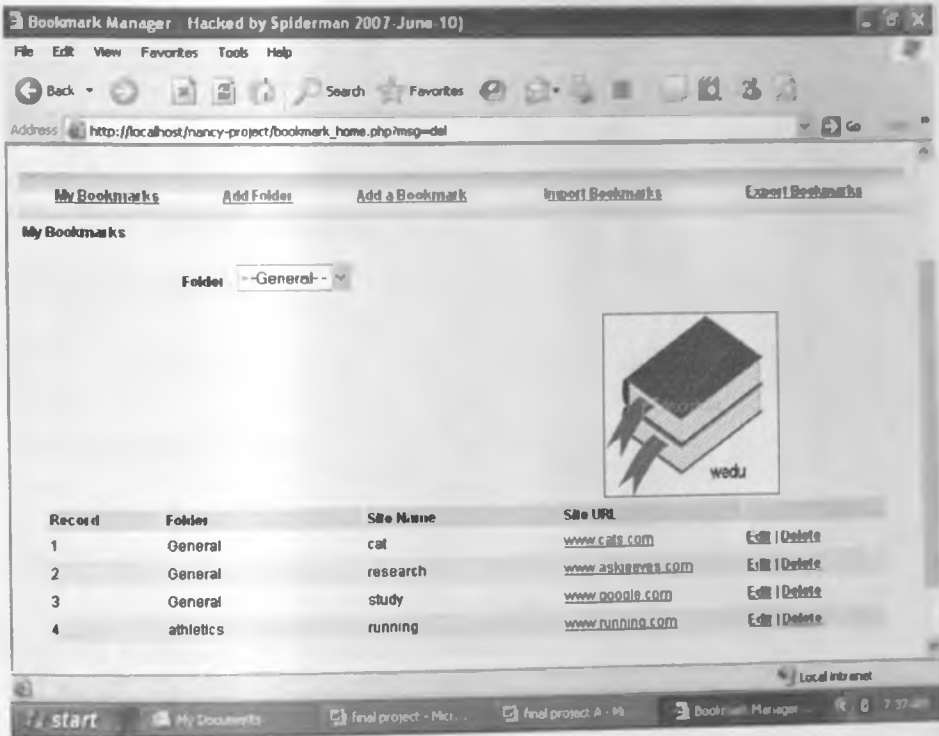
If not registered



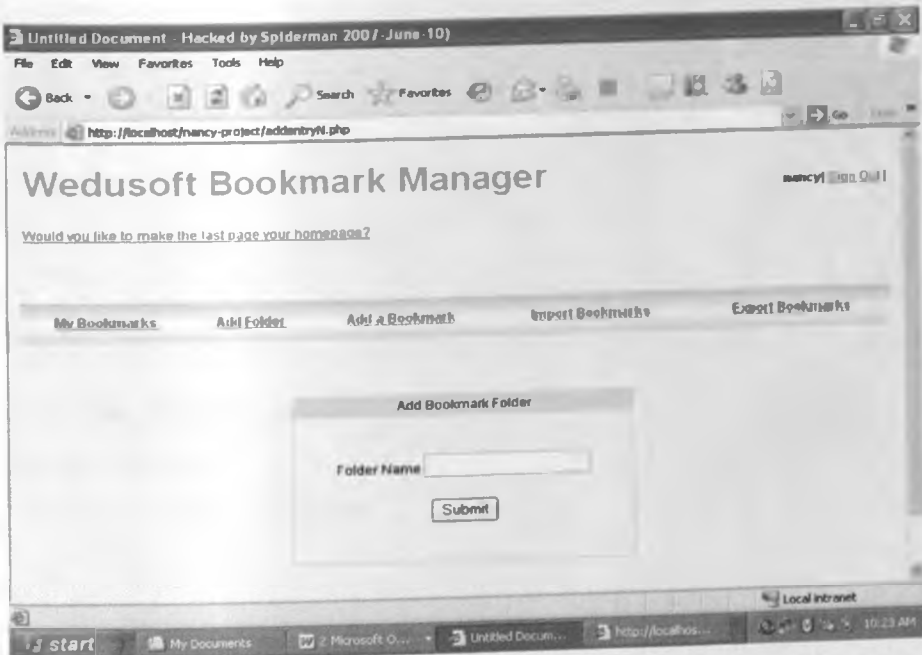
Successful subscription



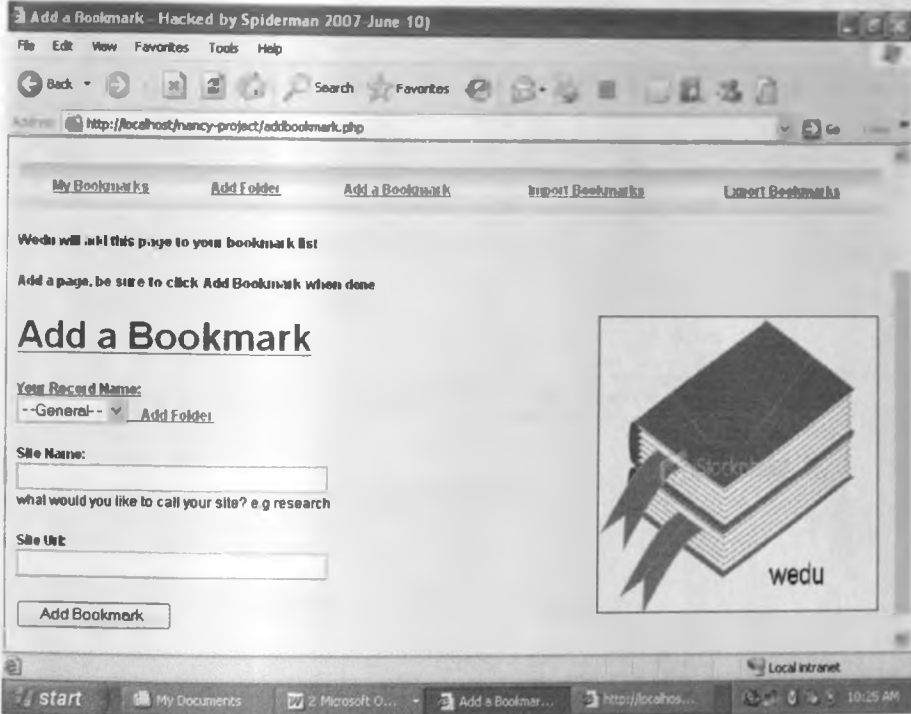
Home Page



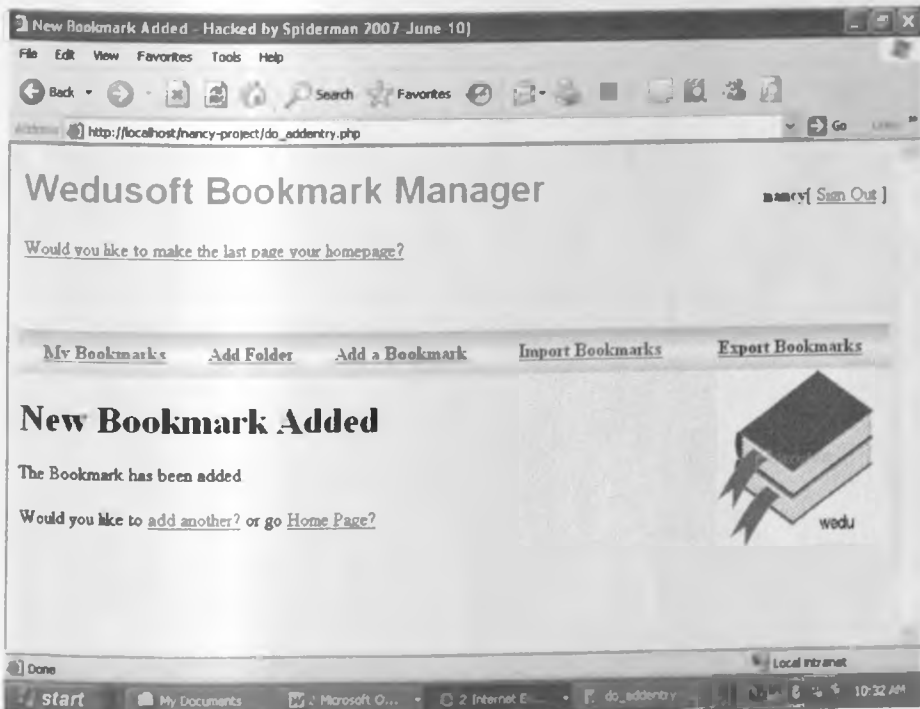
Adding a folder



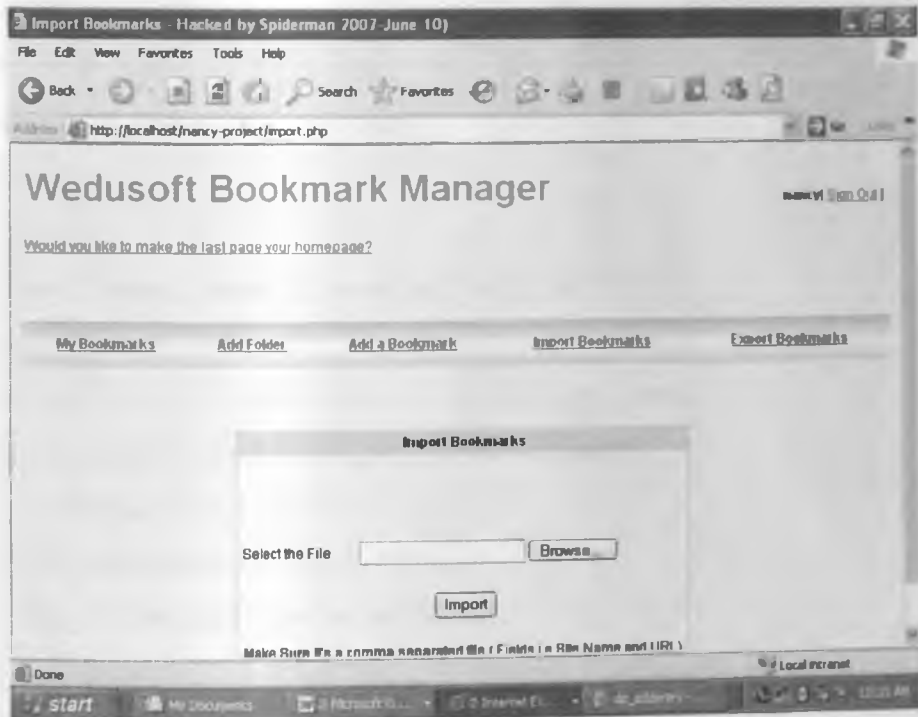
Add a Bookmark



Successful bookmark addition



Import bookmark



7.3 APPENDIX C: SAMPLE CODE

0. PHP code for creating tables

User_name table (user_table.php)

```
<?php
$conn = mysql_connect("localhost","root","nje55ri");
mysql_select_db("testDB" ,$conn);
$sql="CREATE TABLE user_name (
  user_name_id int(11) NOT NULL auto_increment,
  userid mediumint(5) NOT NULL,
  date_added datetime default NULL,
  date_modified datetime default NULL,
  email varchar(150) default '0',
  PRIMARY KEY (user_name_id),
  KEY userid (userid)";
$result = mysql_query($sql, $conn) or die(mysql_error());
echo $result;
?>
```

Url_of_site (site_url.php)

```
<?php
$conn = mysql_connect("localhost","root","nje55ri");
mysql_select_db("testDB" ,$conn);
$sql=" CREATE TABLE url_of_site (
  url_of_site_id int(11) NOT NULL auto_increment,
  user_name_id int(11) NOT NULL default '0',
  date_added datetime default NULL,
```

```
date_modified datetime default NULL,  
name_of_site varchar(50) NOT NULL,  
siteurl varchar(150) default NULL,  
folder varchar(150) default '0',  
PRIMARY KEY (url_of_site_id)  
);  
$result = mysql_query($sql, $conn) or die(mysql_error());  
echo $result;  
?>
```

1. Manage.php (subscription)

```
<?php  
  
$host="localhost"; // Host name  
$username="root"; // Mysql username  
$password="nje55ri"; // Mysql password  
$db_name="testdb"; // Database name  
$tbl_name="user_name"; // Table name  
  
?>
```

2. 0 addbookmark.php

```
<?php  
include_once("db_lib.php");  
include_once("session_lib.php");  
  
if(!is_in_session())
```

```
{
    header("Location:login.php");
    exit;
}
include("panel.php");
?>
<html>
<head>
<title>Add a Bookmark</title>

<script>

function validate()
{
if (document.addentry.email.value=="")
{
alert("please input the bookmark name.")
return false
}
if (document.addentry.sitename.value=="")
{
alert("please input the sitename.")
return false
}
if (document.addentry.siteurl.value=="")
{
alert("please input the siteurl.")
return false
}
}
}
</script>
```

```

<link href="bookmark.css" rel="stylesheet" type="text/css">
</head>
<body BGCOLOR="#FFFFCC">
<P><STRONG>Wedu will add this page to your bookmark list</STRONG>

    <P><STRONG>Add a page, be sure to click Add Bookmark when done
</STRONG></P>

<table border="0" width="100%"><a href="manage.php"><IMG SRC="wedu2.bmp"
ALIGN="right"
alt="Wedusoft Bookmark Manager"
back to subscription form
</a>

<h1>Add a Bookmark</h1>
<form name="addentry" method="post"
ONSUBMIT="return validate()"
action="do_addentry.php">
<p><strong>Your Record Name:</strong><br>
<select name="email">
<option value="0" selected>- General- </option>
    <?php
        $conn = mysql_connect($host, $username, $password) or die(mysql_error());
        mysql_select_db("testDB", $conn) or die(mysql_error());

        session_start();
        $userid=$_SESSION['USERID'];

        $sql="SELECT * FROM user_name WHERE userid='$userid'";

```



```

//connect to server and select database
$conn = mysql_connect($host, $username, $password) or die(mysql_error());
mysql_select_db("testDB",$conn) or die(mysql_error());

//add to user_name table
    $add_home= "update url_of_site SET name_of_site='$_POST[sitename]',
siteurl='$_POST[siteurl]' WHERE url_of_site_id='$_REQUEST[urlid]";
    mysql_query($add_home,$conn) or die(mysql_error());

    header("Location: bookmark_home.php?msg=upd");
    exit;

}
?>
<html>
<script>

function validate()
{
if (document.addentry.siteurl.value=="")
{
alert("please input the siteurl.")
return false
}
}
</script>
<head>

<title>Last Visited Page</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

```

```

<link href="bookmark.css" rel="stylesheet" type="text/css">
</head>

<body BGCOLOR="#ffffcc">
<?php
include("panel.php");
?>
<p>&nbsp;</p>
<?php

$conn = mysql_connect($host, $username, $password) or die(mysql_error());
mysql_select_db("testDB",$conn) or die(mysql_error());

$sql="SELECT * FROM url_of_site WHERE url_of_site_id='$_REQUEST[urlid]'";
$result=mysql_query($sql,$conn)or die(mysql_error());
$info=mysql_fetch_array($result);
$sitename=$info['name_of_site'];
$siteurl=$info['siteurl'];
?>
<table width="400" border="0" align="center" bgcolor="#FFDE9B">
<tr>
<td>
<div align="center"><strong>Last Visited Page</strong></div></td>
</tr>
<tr>
<td bgcolor="#ffffcc">
<form name="addentry" method="post" ONSUBMIT="return validate()" action="">
<table width="100%" border="0">
<tr>
<td>Site Name</td>

```

```

        <td><input name="sitename" type="text" id="sitename" value="<?php echo
        $sitename; ?>"></td>
    </tr>
    <tr>
        <td>Site URL</td>
        <td><input name="siteurl" type="text" id="siteurl" value="<?php echo $siteurl;
        ?>"></td>
    </tr>
</table>
<p align="center">
    <input name="urlid" type="hidden" id="urlid" value="<?php echo
    $_REQUEST['urlid']; ?>">
    <input type="submit" name="Submit" value="Update">
</p>
</form></td>
</tr>
</table>
</body>
</html>

```

4.0 Delete Bookmark Code (delentry.php)

```

<?php
include_once("db_lib.php");
include_once("session_lib.php");

if(!is_in_session)
{
    header("Location: login.php");
    exit;
}

```

```
$urlid=$_REQUEST['urlid'];
```

```
session_start();
```

```
$userid=$_SESSION['USERID'];
```

```
$conn = mysql_connect($host, $username, $password) or die(mysql_error());  
mysql_select_db("testDB",$conn) or die(mysql_error());
```

```
$sql="DELETE FROM url_of_site WHERE user_name_id='$userid' AND  
url_of_site_id='$urlid'";
```

```
$result=mysql_query($sql,$conn)or die(mysql_error());
```

```
header("Location: bookmark_home.php?msg=del");  
exit;
```

```
?>
```

5.0 Adding a folder

```
<?php
```

```
include_once("db_lib.php");
```

```
include_once("session_lib.php");
```

```
if($_POST)
```

```
{
```

```
    session_start();
```

```
    $userid=$_SESSION['USERID'];
```

```

//connect to server and select database
$conn = mysql_connect($host, $username, $password) or die(mysql_error());
mysql_select_db("testDB",$conn) or die(mysql_error());

//add to user_name table

$sql= "INSERT INTO user_name (userid, date_added, date_modified, email)
VALUES ('$userid',NOW(),NOW(),$_POST[folder])";
mysql_query($sql,$conn) or die(mysql_error());

header("Location: bookmark_home.php?msg=flid");
exit;

}

?>
<html>
<script>
function validate()
{
    if (document.form1.folder.value=="")
    {
        alert("please input the folder name.")
        return false
    }
}
</script>
<head>
<title>Untitled Document</title>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="bookmark.css" rel="stylesheet" type="text/css">
</head>

<body>
<?php
include("panel.php");
?>
<form name="form1" method="post" action="" ONSUBMIT="return validate(">
<p>&nbsp;</p>
<table width="300" border="0" align="center" bgcolor="#FFDE9B">
<tr>
<td><div align="center">Add Bookmark Folder</div></td>
</tr>
<tr>
<td bgcolor="#ffffcc"><p align="center">&nbsp;</p>
<p align="center">Folder Name
<input name="folder" type="text" id="folder">
</p>
<p align="center">
<input type="submit" name="Submit" value="Submit">
</p>
<p align="center">&nbsp;</p></td>
</tr>
</table>
<p align="center">&nbsp;</p>
</form>
<p>&nbsp;</p>
</body>
</html>

```