

Recommendation system for Web Based Library Management

A.D.M Abeysekara

10/8551 D

MSCIT/09/014



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Faculty of Information Technology

University of Moratuwa

February 2015

Recommendation system for Web Based Library Management



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
A.D.M. Abeysekara
www.lib.mrt.ac.lk

10/8551 D

MSCIT/09/014

Supervised by: Mr. Saminda Premaratne

February 2015

This dissertation is submitted in partial fulfillment of the
requirement of the Degree of MSc in Information Technology

of

the University of Moratuwa

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree and to the best of my knowledge and it does not contain any material previously published or written by another person or myself except where due reference is made in text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations.

Name of Student:

Signature of Student

A.D.M Abeysekara

.....

Date :



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Supervised by:

Signature of Supervisor

Mr. Saminda Premaratne

.....

Date :

Acknowledgement

I would never have been able to finish my dissertation without the support of Director capital market education and training, and the members of the SEC Library staff. I would like to express my deepest gratitude to my supervisor, Mr. Saminda Premaratne, senior lecturer faculty of information technology university of Moratuwa for his excellent guidance, patience, and providing me with an excellent sharing of knowledge for doing research. And also I need to thank all my family members who were always there cheering me up and stood by me through the good times and bad.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Abstract

The Securities and Exchange Commission of Sri Lanka (SEC) was established in pursuance of the Securities and Exchange Commission of Sri Lanka Act, No. 36 of 1987 as amended by Act No. 26 of 1991, Act No. 18 of 2003 and Act no. 47 of 2009. The Securities and Exchange Commission (SEC) is the regulator for securities market in Sri Lanka. There are several objectives of the SEC such as the creation and maintenance of a market in which securities can be issued and traded in an orderly and fair manner, the protection of the interest of investors, the operation of a compensation fund to protect investors from financial loss arising as a result of any licensed stock broker or licensed stock dealer being found incapable of meeting his contractual obligations; and the regulation of the securities market and to ensure that professional standards are maintained in such a market.

The project is mainly focused in the development of a web based software solution to streamline the operations in the SEC's library. The SEC library consists of more than 1500 books in more than 10 categories with over 100 members. The system will help the librarian to perform basic library functions and allow members to reserve books through it. The major component of the system is its recommendation facility which will give suggestions for the users to select books based on their preferences.



Contents

	Page
Chapter 1 - Introduction	1
1.1 Introduction	1
1.2 Background and Motivation	2
1.3 Problem in Brief	3
1.4 Aim	4
1.5 Objectives	4
1.6 Proposed Solution	4
1.7 Structure of the thesis	4
Chapter 2 - Literature Review	6
2.1 Introduction	6
2.2 Major Approaches in Recommender Systems	8
2.3 Collaborative filtering	8
2.4 Content-based filtering	8
2.5 Model – View – Controller (MVC) Architecture	8
2.5.1 The Model	9
2.5.2 The View	9
2.5.3 The Controller	10
2.6 MVC in Proposed solution	10
2.7 Similar Solutions	10
2.7.1 BibTip	10
2.7.2 ExLibris bX	11
2.7.3 Foxtrot	11
2.7.4 TechLens	12
2.7.5 Fab	12
2.7.6 LIBRA	13
2.8 Summary	13

Chapter 3 - Technology Adapted	14
3.1 Introduction	14
3.2 WAMP	14
3.3 PHP	14
3.4 MySQL	15
3.5 APACHE	15
3.6 Java Scripts	16
3.7 HTML5	16
3.8 CodeIgniter	16
3.9 Windows Operating System	16
3.10 Summary	17
Chapter 4 - Approach	18
4.1 Introduction	18
4.2 Measurements	20
4.3 Approaches in Analysis and Designing	21
4.4 System Users	22
4.5 Inputs, Processes and Outputs	23
4.6 Summary	24
Chapter 5 - Analysis and Design	25
5.1 Introduction	25
5.2 Design of the proposed solution	28
5.3 ER Diagram	29
5.4 Use Case Diagram	31
5.5 Database Implementation	32
5.6 Summary	35
Chapter 6 - Implementation	36
6.1 Introduction	36
6.2 Recommendations on most viewed books	36
6.3 Recommendations on reservation related books	37



6.4 Recommendations on most preferred books	37
6.5 Search Books	38
6.6 View users	38
6.7 Reservation Report	39
6.8 All Books Report	39
6.9 Book reservation	40
6.10 My Reservations	40
6.11 Summary	41
Chapter 7 - Evaluation	42
7.1 Introduction	42
7.2 Black Box Testing	42
7.3 User Interface Testing	44
7.4 Functional testing	45
7.5 Acceptance testing	46
7.6 Evaluate System	46
7.7 Usability Evaluation	49
7.8 Summary	50
Chapter 8 - Conclusion and Further Work	51
8.1 Introduction	51
8.2 Achievement of the Aim and the Objectives of the project	51
8.3 Challenges of the project and Problems encountered	51
8.5 Summary	52
References	53
Appendix A : Main Controller of the System	54
Appendix B : Screens of the System	64



List of Tables

	Page
Table 5.1: Phases of Systems Development Life Cycle	29
Table 5.2: Structure of the table ‘tbl_authors’	33
Table 5.3: Structure of the table ‘tbl_books’	33
Table 5.4: Structure of the table ‘tbl_category’	34
Table 5.5: Structure of the table ‘tbl_publisher’	34
Table 5.6: Structure of the table ‘tbl_recommendations’	34
Table 5.7: Structure of the table ‘tbl_reservations’	34
Table 5.8: Structure of the table ‘tbl_users’	35
Table 5.9: Structure of the table tbl_book_preferences	35
Table 7.1: Black Box Testing	47
Table 7.2: User Interface Testing	49
Table 7.3: Functional Testing	52
Table 7.4: Quality Assurance Testing	52



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

List of Figures

	Page
Figure 2.1: Graphical representation of MVC architecture	9
Figure 4.1: Spiral Model for Recommendation System	22
Figure 4.2: Recommendation System	23
Figure 5.1: Proposed Solution	28
Figure 5.2: ER Diagram	30
Figure 5.3: Use Case Diagram	31
Figure 5.4: Database Implementation	35
Figure 6.1: Recommendations based on most viewed books	36
Figure 6.2: Recommendations based on reservation related books	37
Figure 6.3: Recommendations based on most preferred books	37
Figure 6.4: Search Books	38
Figure 6.5: View all the users of the system	38
Figure 6.6: Reservation Report	39
Figure 6.7: All Books Report	39
Figure 6.8: View book reservations	40
Figure 6.9: My Reservations	40



Chapter 1

Introduction

1.1 Introduction

Library automation is a discipline in information and communications technologies (ICT) that are used to replace manual systems in the library. Reservation of books online is an important sub function in the library management system which helps users to select books or check availability by logging through a remote computer without being physically available in the library. Reservation of books can be done mainly in two ways, depending on the decision of the user or based on the recommendation of the book reservation system.

Recommender systems or recommendation systems are software tools and techniques providing suggestions for items to be of use to a user. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to or what online news to read. “Item” is the general term used to denote what the system recommends to users. A Recommender System normally focuses on a specific type of item and accordingly its design, its graphical user interface, and the core recommendation technique which are used to generate the recommendations are all customized to provide useful and effective suggestions for that specific type of item. Recommender systems are primarily directed towards individuals who lack sufficient personal experience or competence to evaluate the potentially overwhelming number of alternative items that a website, for example, may offer. A case in point is a book recommender system that assists users to select a book to read. In the popular website, Amazon.com, the site employs a recommendation system to personalize the online store for each customer. Here the recommendation feature of the system, suggests to the users to reserve books based on their past reservation patterns, which is also known as collaborative filtering.

1.2 Background and Motivation

The Securities and Exchange Commission of Sri Lanka (SEC) was established in pursuance of the Securities and Exchange Commission of Sri Lanka Act, No. 36 of 1987 as amended by

Act No. 26 of 1991, Act No. 18 of 2003 and Act no. 47 of 2009. Securities and Exchange Commission (SEC) is the regulator for securities market in Sri Lanka. There are several objectives of SEC such as the creation and maintenance of a market in which securities can be issued and traded in an orderly and fair manner, the protection of the interest of investors, the operation of a compensation fund to protect investors from financial loss arising as a result of any licensed stock broker or licensed stock dealer being found incapable of meeting his contractual obligations; and the regulation of the securities market and to ensure that professional standards are maintained in such market.

SEC comprises with 9 divisions consisting of more than 150 employees, the said 9 divisions are Capital Market Development Division, Capital Market Education and Research Division, Capital Market Surveillance Division, External Relations Division, Supervision Division, Finance and Administration Division, Investigation Division, Legal Division and Corporate Affairs Division.

Except the support staff SEC's executive and non-executive staffs are undergraduates and 80% of them have completed their post graduate qualifications, so the usage of SEC's Library is quite high. In addition to that the capital market research is extensively carried out by SEC staff, which in turns increases the library operations at SEC. It is believed that by the automation of the Library functions at SEC will increase efficiency and solve an array of erroneous practices.

1.3 Problem in Brief

In the current context all the operations in SEC's library are conducted manually which may leads to several problems.

Data duplication can occur by repeating the same thing over and over. Lack of security, lack of storage common errors. Too much paper wastage. Paper takes up a massive amount of room in the site. If a computer system was bought the paper could all go and there would lots more free space which could be used for other purposes. Poor Data Storage - All the data is stored in filing cabinets. Data could be misplaced due to human error. Data could be stolen very easily. Unavailability of Information· Slow Retrieval of Data - The information is stored

in different parts and takes a long time to retrieve the data. In a manual library system if a member is not able to find a book of his / her choice have to spend a lot of time and effort in trying to find a particular book. Manual methods are also slow to operate. Instead of using a computer to issue and take back books, locating and updating a card index is slow and laborious.

Librarian finds it difficult to offer a wider range of new services with a manual library system. For example, a library can put its catalog on the Web allowing readers to access it remotely; with a manual system members have to visit or telephone the library to find this information.

Not more than one person can access data at the same time but in Computerized Information system many people can access the same data on the same time through networking. Often the books are lost and the librarian is not aware of this. No proper records for the members and book transactions.

If manual record book is lost data will be completely lost. Manual labor is required for record keeping. Many a times duplication occurs as workers are find it hard to keep track in the bundles of registers. Data is not always reliable as it is hand written and some human errors might have occurred eg:- wrong telephone number.

1.4 Aim

The aim of this project is to develop a system to automate the library operations and a book recommendation system at SEC through web technologies.

1.5 Objectives

- Incorporate collaborative filtering mechanism to the library system.
- Make the system automatically recommend library members to reserve books based on their preferences.
- Automate the library functions of the SEC, which are conducting manually at present.
- Reduction of manual work in the library functions which will cause to reduce the errors in the process.

1.6 Proposed Solution

Proposed solution would enable members of the library to reserve books online and also enable librarian to maintain book information in a web based system and makes the functions of the library more flexible and accurate.

1.7 Structure of the thesis

Chapter 1 provides a high level synopsis on the project undertaken. Chapter 2 outlines the underlying theories of the project. Chapter 3 Outlines the technologies adopted and provides the architecture of the system developed. Chapter 4 describes project approach adopted to develop the system. The Spiral Model in software development has been highlighted as the development methodology. In chapter 5 system analysis and design approach has been described. Techniques used in requirements gathering, analysis and design, implementation and post implementation review as been critically evaluated in this chapter. In chapter 6 the final working system has been described as an operational guide. Chapter 7 describes all evaluation techniques used including various testing and quality assurance methodologies adopted in the project. Finally chapter 8 gives an overall evaluation with the achievements, gaps and further improvements which could be adopted in future.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Chapter 2

Literature Review

2.1 Introduction

IT plays a pivotal role in each and every field in today's world. It enhances operational and functional side of each and every business. Library management systems are one of the major fields which will be helpful for librarians and its members to use and manage library facilities and its infrastructure.

Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user. The suggestions provided are aimed at supporting their users in various decision-making processes, such as what items to buy, what music to listen, or what news to read. Recommender systems have proven to be valuable means for online users to cope with the information overload and have correspondingly, various techniques for recommendation generation have been proposed and during the last decade, many of them have also been successfully deployed in commercial environments.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

RSs are information processing systems that actively gather various kinds of data in order to build their recommendations. Data is primarily about the items to suggest and the users who will receive these recommendations. But, since the data and knowledge sources available for recommender systems can be very diverse, ultimately, whether they can be exploited or not depends on the recommendation.

Items are the objects that are recommended. Items may be characterized by their complexity and their value or utility. The value of an item may be positive if the item is useful for the user or negative if the item is not appropriate and the user made a wrong decision when selecting it.

Users of a RS, as mentioned above, may have very diverse goals and characteristics. In order to personalize the recommendations and the human-computer interaction, RSs exploit a range of information about the users. This information can be structured in various ways and again the selection of what information to model depends on the recommendation technique.

The problem of recommending items has been studied extensively, and two main paradigms have emerged. *Content-based* recommendation systems try to recommend items similar to those a given user has liked in the past, whereas systems designed according to the *collaborative* recommendation paradigm identify users whose preferences are similar to those of the given user and recommend items they have liked

The abundance of information available on the Web and in Digital Libraries, in combination with their dynamic and heterogeneous nature, has determined a rapidly increasing difficulty in finding what we want when we need it and in a manner which best meets our requirements.

As a consequence, the role of user modeling and personalized information access is becoming crucial: users need a personalized support in sifting through large amounts of available information, according to their interests and tastes.

Recommender systems have the effect of guiding users in a personalized way to interesting objects in a large space of possible options. *Content-based* recommendation systems try to recommend items similar to those a given user has liked in the past. Indeed, the basic process performed by a content-based recommender consists in matching up the attributes of a user profile in which preferences and interests are stored, with the attributes of a content object (item), in order to recommend to the user new interesting items.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

2.2 Major Approaches in Recommender Systems

Most recommender systems take either of two basic approaches: collaborative filtering or content-based filtering (Jones, 2013).

2.3 Collaborative filtering

Collaborative filtering arrives at a recommendation that's based on a model of prior user behavior (Jones, 2013). The model can be constructed solely from a single user's behavior or more effectively also from the behavior of other users who have similar traits. When it takes other users' behavior into account, collaborative filtering uses group knowledge to form a recommendation based on like users. In essence, recommendations are based on an automatic collaboration of multiple users and filtered on those who exhibit similar preferences or behaviors.

2.4 Content-based filtering

Content-based filtering constructs a recommendation on the basis of a user's behavior. (Jones, 2013) For example, this approach might use historical browsing information, such as which blogs the user reads and the characteristics of those blogs. If a user commonly reads articles about Linux or is likely to leave comments on blogs about software engineering, content-based filtering can use this history to identify and recommend similar content (articles on Linux or other blogs about software engineering). This content can be manually defined or automatically extracted based on other similarity methods.

2.5 Model – View – Controller (MVC) Architecture

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. (tutorialspoint) A Model View Controller pattern is made up of the following three parts. MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here, the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response.

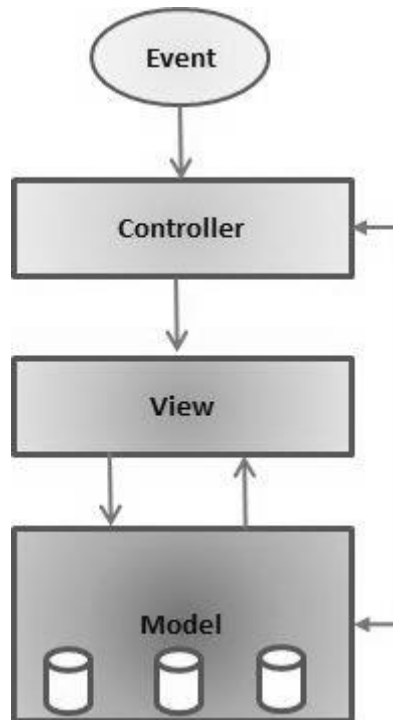


Figure 2.1 – graphical representation of MVC architecture

2.5.1 The Model

The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.



2.5.2 The View

A presentation of data in a particular format, triggered by a controller's decision to present the data. They are script based templating systems like JSP, ASP, PHP and very easy to integrate with AJAX technology.

2.5.3 The Controller

The controller is responsible for responding to user input and perform interactions on the data model objects. The controller receives the input, it validates the input and then performs the business operation that modifies the state of the data model.

2.6 MVC in Proposed solution

The recommendation system for the library management system is also based on MVC architecture and Due to the separation of the model from the view; the user interface can display multiple views of the same data at the same time, which we can call as multiple view support. In the meantime User interfaces tend to change more frequently than business rules. (different colors, fonts, screen layouts, and levels) Because the model does not depend on the views, adding new types of views to the system generally does not affect the model. As a result, the scope of change is confined to the view.

2.7 Similar Solutions

At present there are several recommendation systems for library management are available as follows.

2.7.1 BibTip

Since 2009 BibTip has been a paid service of a spin-off from the University of Karlsruhe. (tutorialspoint) It performs a behavior based RS that generates recommendations based on an anonymous monitoring of user behavior and its evaluation. The data analysis and management of the recommendations take place on the servers of BibTip. The libraries themselves need only a small technical effort to integrate the service. BibTip follows three-level architecture with data on the lowest level, agents on the middle level and user interface on the top level. BibTip is based on the repeat-buying theory by Andrew Ehrenberg, which analyzes consumer behavior. Ehrenberg proved that people repeat their choices after a purchase decision and trust the same brand next time. BibTip makes use of this theory. The users' \initial" behavior is used to draw conclusions about interests. Example: If the user is interested in Publication X, he is also interested in publication Y by the same author. BibTip requires a certain initial period up to several months in which the behavioral data is collected and analyzed. A highly frequented BibTip system can reduce the start-up phase. But still this RS suffers from the cold start problem. Besides, BibTip can be applied to any resources, as it generates its recommendations from user behavior.

2.7.2 ExLibris bX

Since 2009 a library can easily activate bX as an on-demand service (but only if it is SFX customer). bX is a research result by the authors of (Francesco Ricci • Lior Rokach • Bracha Shapira). bX is based on standards such as OpenURL and OAI-PMH, two popular protocols for unique identification of resources and related metadata. bX as behavior-based RS is a system that generates recommendations based on the analysis of clicked links within a web session. The necessary statistics are collected by the ExLibris Link Resolver in log files. Log files from all available institutions are used to generate recommendations. bX tries to mitigate problems of conventional behavior-based RS. This includes simple approaches to the de-duplication of publications and users. The cold start problem is obviously low, as log data from many institutions are used. If there are no other institutions, bX must collect enough data first.

2.7.3 Foxtrot

Foxtrot uses ontology (research paper topic ontology) to describe the user and also builds profiles from it. (tutorialspoint) Foxtrot generates recommendations based on similar users and similar resources. Thus Foxtrot is a hybrid RS as a mixture of collaborative and content-based approaches. In addition, the interactions (visited URLs, feedback) with the user are logged through a web proxy. A visualized profile which consists of popular publications and dates allows the user to understand how the system works. Each of the available paper is described by a vector of normalized terms together with metadata such as date, title, classification, links, and the paper's URL. Foxtrot uses the multi-class classifier IBK 2 to classify the papers. The profiles which represent the interests of users are enhanced with ontological inference. Since the ontology is a simple IS-A taxonomy the possibilities of inferred knowledge are small. But Foxtrot exploits subclass relationships. The user profile is extended by the generalized super topic for each topic. Foxtrot also offers the possibility to integrate an external ontology with relevant data, thus eliminating the cold start problem.

2.7.4 TechLens

The most important feature of TechLens is that its collaborative method focuses on the paper themselves (tutorialspoint). Thus, a paper implicitly evaluates its references as good papers, which will be used for recommendations. The authors mention some profile possibilities and

decided to use the "One-Paper-Explicit-Short-Term" variant. The profile consists only of one single paper (implicitly: the last seen one, explicitly: user selects this paper). As to authors of this matches the current users' interest best. The advantage of this profile is that no other monitoring system is necessary on the user side. The disadvantage is that an interest development can't be observed. 10 different collaborative, content-based and hybrid algorithms have been presented. The extensive test results show that the hybrid approach "Fusion" produced qualitatively better results than a pure collaborative method. Fusion uses both approaches (collaborative and content-based) which run in parallel. Each recommendation that is found in both lists is added to the result list with a particular ranking. Besides, the implemented user profile is not considered suitable in the long term, as a single paper brings a very limited knowledge capacity.

2.7.5 Fab

Fab, which was developed within the Digital Library Project at Stanford University is also a hybrid RS combining collaborative filtering with content-based methods (Jones, 2013). Designed in 1997 before the major search engines FAB has the task to select relevant and interesting websites from the mass of available websites. Recommendations are based on ratings already given by the active user and those of other similar users. Users are described in their profile with a weighted term vector. The updating of the profiles is done with relevance feedback using the Rocchio algorithm. The web sites are also represented by a weighted term vector of dimension 100, resulting from the application of TF-IDF.

2.7.6 LIBRA

LIBRA stands for "Learning Intelligent Book Recommending Agent". It was specifically designed for digital libraries for the recommendation of books (Gottwald, 2005).. The user must evaluate the first ten search results. Then the result set is sorted according to the preferences of the user. Resources are described by the following attributes such as title, author, summary, reviews, customer reviews, related authors, related titles and keywords. To create a profile the user rates ten books with values of 1 (poor) to 10 (good). The learning algorithm Naive Bayes from the field of machine learning is used to generate a ranking of the books for the user. To generate recommendations LIBRA searches for users similar to the active user. The similarity of the users is calculated using Pearson correlation. Then rating prediction is made based on the ratings of similar users

2.8 Summary

The chapter describes field of recommender systems. And different approaches in recommendation system. Chapter describes MVC architecture as the basis for the proposed solution and also it describes similar products which are blended with recommender systems. One of the major limitations of the above mentioned systems is that recommendation mechanism is adopted only on one specific criteria by focusing only on customer ratings of books, unlike in e-commerce sites, in the library systems member ratings of books are rarely done.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Chapter 3

Technology Adapted

3.1 Introduction

Client-server architecture has been adopted for the Recommendation System. (Chung) This would enable multi user environment across number of platform due to the use of web based products.

Following open source software components have been used for the development of the system.

- WAMP – Server environment simulation
- PHP – Server-side scripting language
- MySQL – Database server environment
- APACHE – Application Server
- HTML 5 and JAVA Scripts – Client-side scripting and programming
- CodeIgniter – Web application framework



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

3.2 WAMP

WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your databases.

3.3 PHP

Development of the Recommendation system has been carried using PHP for server-side scripting requirements. PHP is a server-side scripting language designed for web development purposes. (Group, 2001) PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone graphical applications.

PHP includes free and open source libraries with the core build. PHP is a fundamentally Internet-aware system with modules built in for accessing File Transfer Protocol (FTP) servers, many database servers, embedded SQL libraries such as MySQL, Microsoft SQL Server and others.

PHP is a general-purpose scripting language that is especially suited to server-side web development where PHP generally runs on a web server. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS).

3.4 MySQL

The database of the recommendation system running on MySQL, which is an open source relational database management system that runs as a server providing multi-user access to a number of databases (Chung).

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP and XAMPP open source web application software stack.



University of Moratuwa, Sri Lanka
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

3.5 APACHE

The Recommendation system runs on APACHE server for windows. The Apache HTTP Server, is a web server software program which has the ability run on multiple operating systems. (Gottwald, 2005)

Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Virtual hosting allows one Apache installation to serve many different websites. For example, one machine with one Apache installation could simultaneously serve www.example.com, www.example.org, test47.test-server.example.edu, etc.

Apache features configurable error messages, DBMS-based authentication databases, and content negotiation. It is also supported by several graphical user interfaces (GUIs).

3.6 Java Scripts

JavaScript (JS) is an interpreted computer programming language. It was originally implemented as part of web browsers so that client-side scripts could interact with the user, control the browser, communicate asynchronously, and alter the document content that was displayed.

3.7 HTML5

HTML version 5 has been used HTML5 is a markup language for structuring and presenting content for the World Wide Web and a core technology of the Internet

3.8 CodeIgniter

CodeIgniter is an open source rapid development web application framework, for use in building dynamic web sites with PHP. The first public version of CodeIgniter was released on February 28, 2006, and the latest stable version 2.2.1 was released January 22, 2015.

CodeIgniter is loosely based on the popular Model-View-Controller development pattern. While controller classes are a necessary part of development under CodeIgniter, models and views are optional. CodeIgniter is most often noted for its speed when compared to other PHP frameworks.

3.9 Windows Operating System

The Recommendation system at present has been tested on Windows 7 and Windows XP desktop environment with the use of XAMPP simulation software kit.

3.10 Summary

A client server based application has been developed with the aim of facilitating a multiuser environment for the proposed solution. The entire application has been developed using free and open source software facilitating the initial step to be more affordable. However, in the event the organization wishes to move for proprietary platform the same model could be adopted.

Chapter 4

Approach

4.1 Introduction

It is evident that identification of all stakeholders, comprehensive process analysis, proper system design, use of correct hardware and software, smooth installation, testing, implementation, and post implementation review are key to success of any software project.

In general, Spiral Model would be used to manage the entire project. The Spiral model could be used to transform the existing process into a computers driven system in an organization Recommendation system implementation would follow series of iterations within the spiral model until a solid output is produced. Meanwhile, best practices of project management framework as outlined by the Project Management Institute USA (PMI) too would be following in order to handle the project efficiently.



University of Moratuwa, Sri Lanka.

Electronic Theses & Dissertations

www.lib.mfu.ac.lk

The spiral model is based on continuous refinement of key products for requirements definition and analysis, system and software design, and implementation (the code). At each iteration around the cycle, the products are extensions of an earlier product. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.

Documents are produced when they are required, and the content reflects the information necessary at that point in the process. All documents will not be created at the beginning of the process, nor all at the end (hopefully). Like the product they define, the documents are works in progress. The idea is to have a continuous stream of products produced and available for user review.

The spiral lifecycle model allows for elements of the product to be added in when they become available or known. This assures that there is no conflict with previous requirements and design. This method is consistent with approaches that have multiple software builds and releases and allows for making an orderly transition to a maintenance activity. Another

positive aspect is that the spiral model forces early user involvement in the system development effort.

Starting at the center, each turn around the spiral goes through several task regions.

- Determine the objectives, alternatives, and constraints on the new iteration.
- Evaluate alternatives and identify and resolve risk issues.
- Develop and verify the product for this iteration.
- Plan the next iteration.

Note that the requirements activity takes place in multiple sections and in multiple iterations, just as planning and risk analysis occur in multiple places. Final design, implementation, integration, and test occur in iteration 4. The spiral can be repeated multiple times for multiple builds. Using this method of development, some functionality can be delivered to the user faster than the waterfall method. The spiral method also helps manage risk and uncertainty by allowing multiple decision points and by explicitly admitting that all of anything cannot be known before the subsequent activity starts.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

- Identify Stakeholders
- Analyze the existing environment
- Select methodologies i.e. waterfall model or spiral model.
- Prepare to implement library recommender system framework.
- Decide who will take on the responsibility of book management
- Make sure that management commitment, budget and resource is made available before you consider setting up library recommender system
- Ensure that the proposed solution aligns with your business/organization's strategy and vision
- Define clear objectives and deliverables
- Involve and consult IT staff
- Sell the benefits to the support staff – implementing library recommender system will need a change of behavior from IT staff as well as users
- Plan library recommender system process training
- Decide what reports are required
- library recommender system post implementation review

- User satisfaction analysis and surveys

Satisfaction surveys are an excellent method of monitoring user perception and expectation and can be used as a powerful marketing tool. However, to ensure success you should address several key points:

- Decide on the scope of the survey
- Decide on the target audience
- Clearly define the questions
- Make the survey easy to complete
- Conduct the survey regularly
- Make sure that your users understand the benefits
- Publish the results
- Follow through on survey results
- Translate survey results into actions

4.2 Measurements



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

- Do not set targets that cannot be measured
- Ensure that users are aware of what you are doing, and why
- Establish a baseline before discussing formal Service Level Agreements (SLAs) with customers

4.3 Approaches in Analysis and Designing

As explained in the previous chapter, Spiral Model would be used for the development of Recommendation System for Securities and Exchange Commission of Srilanka.

Following approaches would be adopted in line with the spiral model. The planning would be done in accordance with the best practices outlined by the Project Management Institute.

1. Identify stakeholders and stakeholder responsibilities
2. Collect Requirements
3. Define Scope (Problem Analysis)

4. Create Work breakdown Structures
5. Define activities of the project and develop Schedule
6. Costing and Budgeting
7. Plan for Risk Management
8. Identify Risks
9. Analyze risks and plan risk responses.
10. Execute the project (Software coding
11. Prototype building
12. Perform Quality Assurance and Quality Control
13. Verify deliverables and perform testing
14. Commission the system

Most of the activities listed above would be iterative as the system would be developed as per the spiral model. The advantages would be:

- Changes to the scope could be made as and when new requirements arise.
- Stakeholders engage in the entire systems development life cycle.
- Risks could be identified iteratively and make necessary risk responses.
- Higher the iterative better the quality in the final outcome.



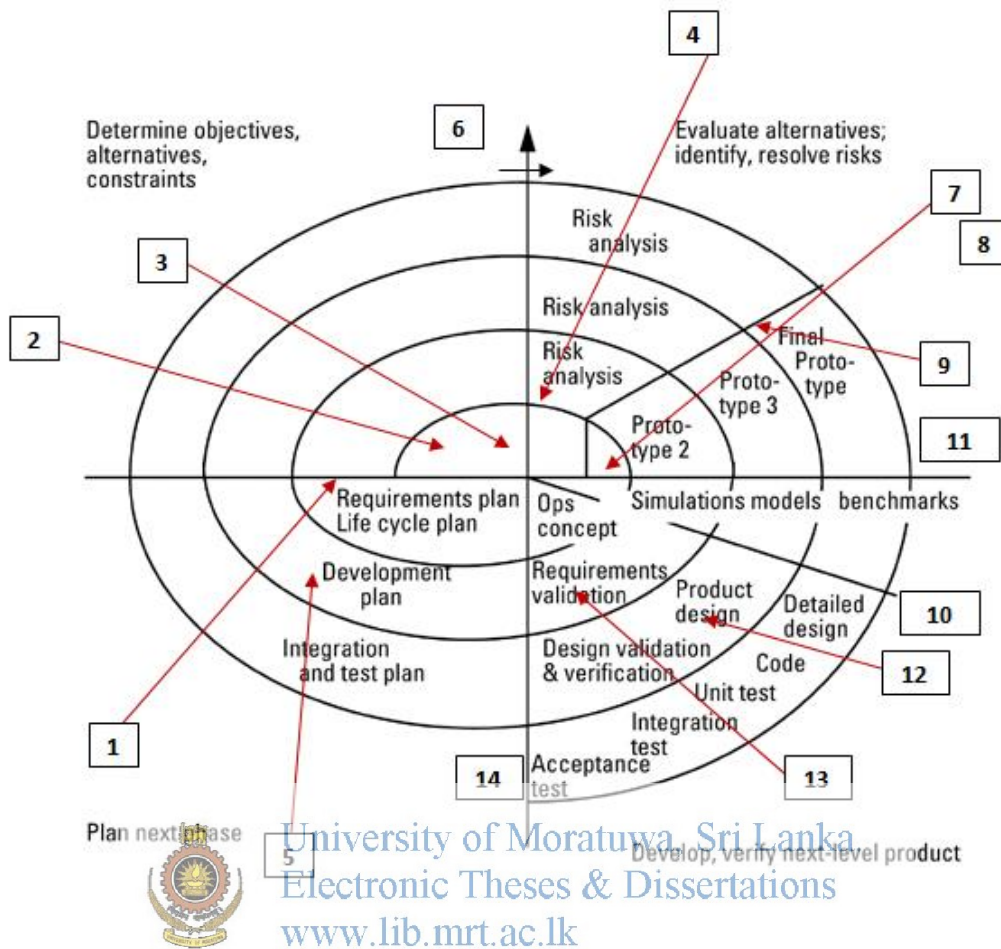


Figure 4.1 : Spiral Model for Recommendation System

4.4 System Users

Following user levels would be defined in the system.

Admin (Librarian) – This is the highest privileged user having access to the entire system. User is responsible for creation of users ,authors, publishers, book categories and books .

Normal Users (Library Members) - members of the library whom can reserve books from the system.

4.5 Inputs, Processes and Outputs

Identification of relevant inputs, processes, outputs and the stakeholder is utmost priority in the system planning process.

Adding books, book categories, publishers, authors are the main set of functions of the book recommendation system which are carried out by the librarian. The core function of the system is to do recommendations for the users to select books based on processing the user's behavior. The recommendation engine of the system is the heart of the system which makes heavy use of book-to-book collaborative filtering approach. This essentially means that for each book X, recommendation engine builds a neighborhood of related books S(X); whenever user reserve/look at a book, system then recommends user books from that book's neighborhood.

So the main input data for the recommendation system are the book reservations and book views of the users. As long as each record captures an interaction between a user and a book, based on those records recommendation engine process and present suggestions to reserve books as the output of the system.

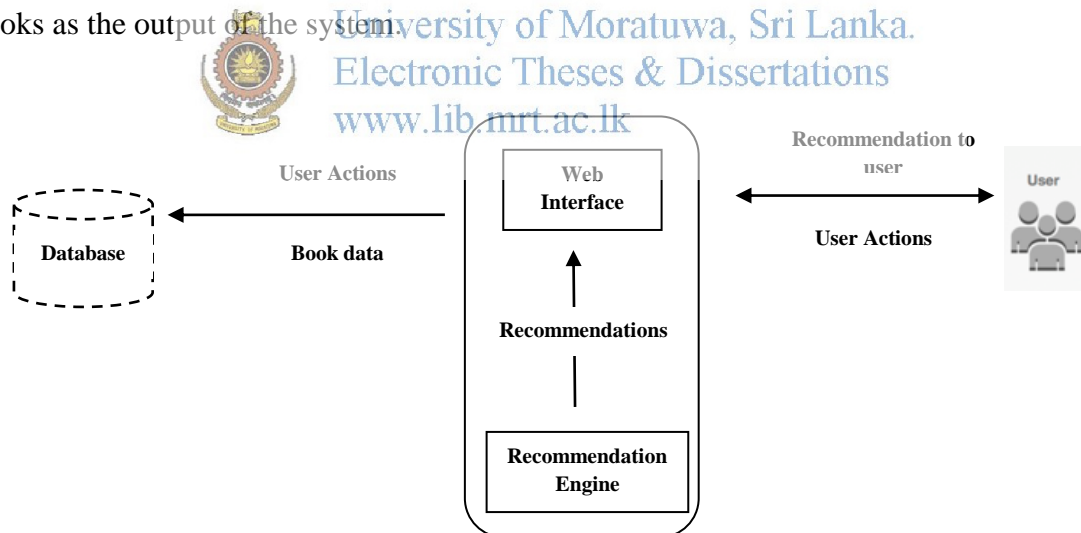


Figure 4.2 : Recommendation System

4.6 Summary

The chapter has explained the steps taken at each of the phase during the systems development life cycle. The spiral model has been adopted due to the continuously changing IT requirement of any organization. As there could be many challenges during the initial phase to collect the requirements, spiral model would facilitate engage in a continuous discussion with all stakeholders. This model allows stakeholders to provide continuous feedback on the model which is being developed, make prototypes and assess the potential working model, and engage in risk assessments. A final design would be signed off after number of iterations which is followed by the testing, implementation and post implementation reviews.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk


Chapter 5

Analysis and Design

5.1 Introduction

The entire project was subject to spiral model where several iterations were done during the process in order to assess the success of the project. The following table outlines the entire workflow which has been conducted in an iterative manner subject to spiral model.

Item	Description	Repetitive processes	Output
Requirements gathering:	Interview all key stakeholders including project owners, sponsors, technical staff and other key decision makers.	<ol style="list-style-type: none"> 1. Determine high-level objectives, alternatives, constraints, and concept. 2. Determine high-level costs, requirements, stakeholder buy-in. 3. Conduct risk analysis 4. Build prototypes 5. Discuss with all stakeholders and evaluate the outcome of prototypes, simulations, and models. 	<p>Repeat this process until Concept of the product is finalized.</p> <p>The main output would be the system requirement specifications, set objectives, design specifications, validated requirements..</p>
Development Plan	A comprehensive project plan must be drawn in order to develop the proposed	<ol style="list-style-type: none"> 1. Gather software requirements 2. Work on product design 	Validated Design and the product

	system.	<p>3. Work on functional and operational prototypes</p> <p>4. Conduct risk management</p> <p>5. Keep all stakeholders informed.</p>	
Integration and testing	<p>Components that are developed in isolation must be integrated and to be tested to ensure the functionality as per the requirement specification.</p> 	<p>1. Perform unit testing,</p> <p>2. Perform integrate testing</p> <p>3. Perform acceptance testing</p> <p>4. Validate against requirements.</p> <p>5. Obtain Sign off</p> <p>6. Conduct risk analysis.</p>	A functional system.
Implementation	A system which has fully tested would be implemented in the production environment.	<p>The implementation must not affect the existing operational environment.</p> <p>If required decide whether the existing manual/computerize system too should run parallel through the new implementation.</p>	A working system
Post implementation	Conduct post implementation review	Obtain feedback from relevant	Documentations, Change request.

Review	in order to evaluate whether the desired expectations have met.	stakeholders.	
--------	---	---------------	--

Table 5.1: Phases of Systems Development Life Cycle

The system analysis was conducted with various stakeholders of the SEC. Following stakeholder inputs were obtained during the analysis process.

- **Ordinary system users:** Users were interviewed and observed in order to obtain their real requirements. Users provided valuable inputs on the features they would require on a recommendation system. Such interviews lead to understand the current scenario, drawbacks and to identify areas needs improvements.
- **Library Staff:** Library staff is the staff which works basically on the library system, which has the first hand experience in library functions. The interviewing results of such staff are a salient input for the system development in terms of library features.
- **IT Staff:** IT staff was interviewed to find out the feasibility of the institution to provide the required infrastructure for the system.
- **Business Sponsors:** Generally, every system must have a sponsor in terms of providing financial and other support when developing a system.

5.3 ER Diagram

Entity–relationship model (ER model) is a data model for describing a database in an abstract way. Following are the entities and related attributes which have been identified for the system. An ER model is an abstract way of describing a database. In the case of a relational database, which stores data in tables, some of the data in these tables point to data in other tables.

- tbl_users
- tbl_books
- tbl_category
- tbl_publishers
- tbl_authors
- tbl_recommendations
- tbl_reservations
- tbl_users
- tbl_book_preferences



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

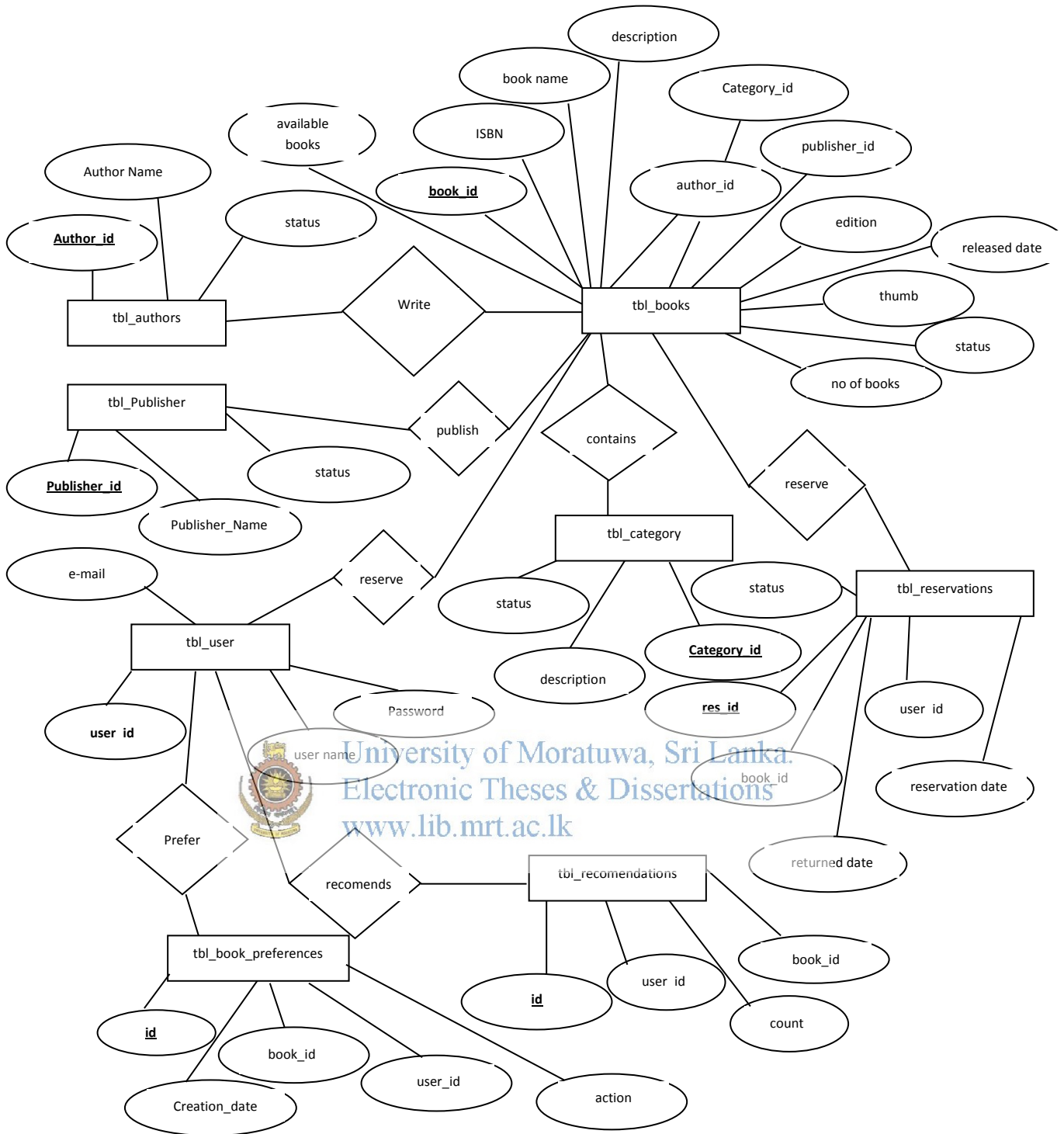


Figure 5.2: ER Diagram

5.4 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different

types of users of a system and the various ways that they interact with the system. The following diagram shows the users and the relevant activities of the entire system.

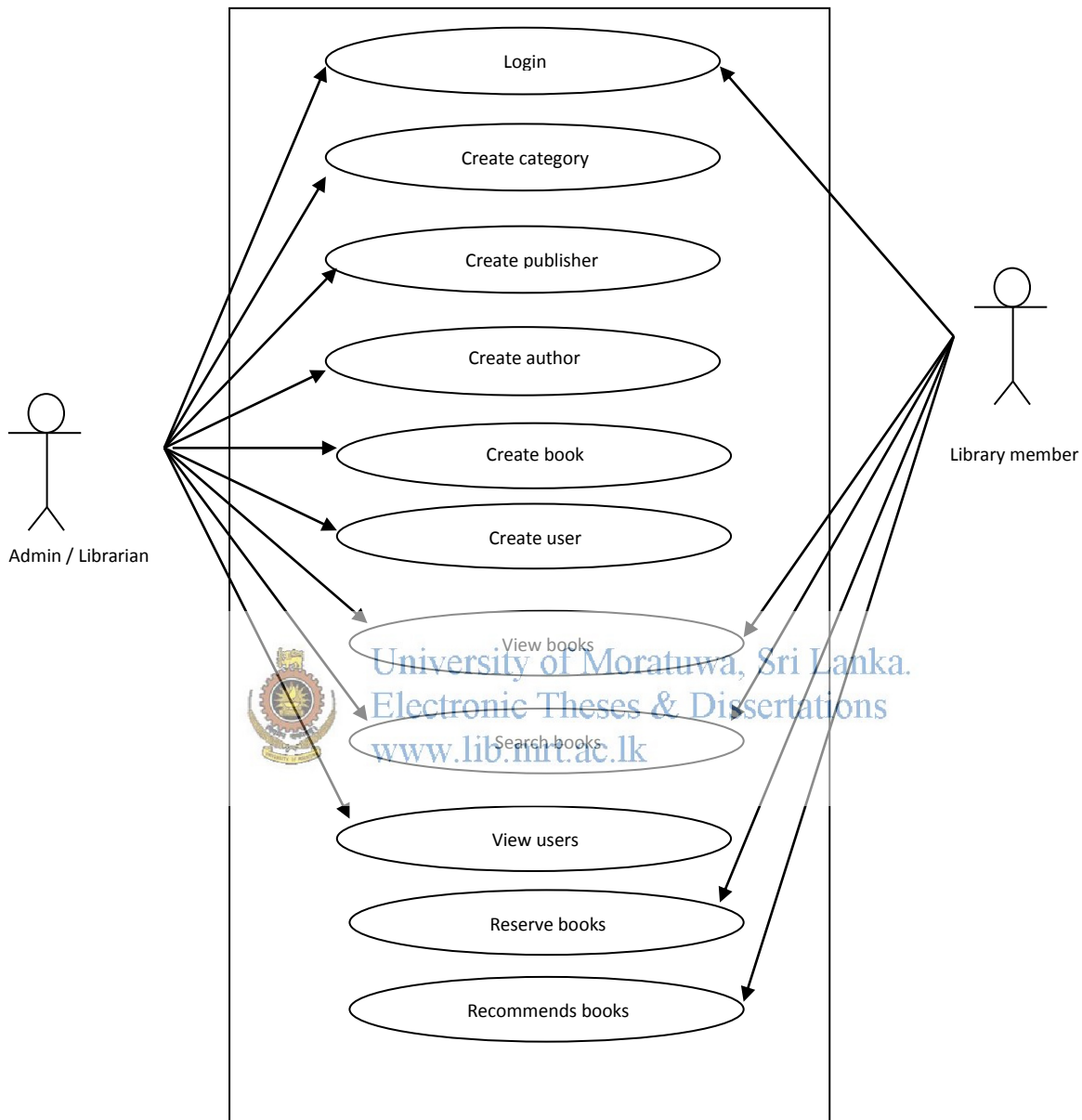


Figure 5.3: Use Case Diagram

5.5 Database Implementation

The database of the application consists of 07 tables. The database diagram shows the relation between tables. The first filed of the table always consider as the primary key.

The structures of tables in the database have been highlighted below.

Column Name	Type	Null	Default
author_id	int(11)	No	
author_name	varchar(250)	No	
Status	enum(0,1)	No	

Table 5.2: Structure of the table 'tbl_authors'

Column Name	Type	Null	Default
book_id	int(11)	No	
ISBN	varchar(100)	No	
book_name	varchar(100)	No	
description	varchar(1000)	No	
cat_id	int(11)	No	
author_id	int(11)	No	
publisher_id	int(11)	No	
Edition	varchar(50)	No	
Thumb	varchar(15)	No	
no_of_books	int(11)	No	
available_books	int(11)	No	
Status	enum('0','1')	No	
Release date	date	No	

Table 5.3: Structure of the table 'tbl_books'

Column Name	Type	Null	Default
cat_id	int(11)	No	
description	varchar(70)	No	
Status	enum(0,1)	No	

Table 5.4: Structure of the table 'tbl_category'

Column Name	Type	Null	Default
publisher_id	int(11)	No	
Publisher name3	varchar(70)	No	

Status	enum(0,1)	No	
--------	-----------	----	--

Table 5.5: Structure of the table 'tbl_publisher'

Column Name	Type	Null	Default
Id	int(11)	No	
user_id	int(11)	No	
book_id	int(11)	No	
Count	int(11)	No	

Table 5.6: Structure of the table tbl_recommendations

Column Name	Type	Null	Default
res_id	int(11)	No	
book_id	int(11)	No	
user_id	int(11)	No	
Count	int(11)	No	
reservation_date	date	No	
returned_date	date	No	
Status	enum('0','1')	No	

Table 5.7: Structure of the table tbl_reservations

Column Name	Type	Null	Default
user_id	int(11)	No	
Username	varchar(20)	No	
Password	varchar(20)	No	
Email	varchar(20)	No	
date_joined	date	No	
user_level	date	No	
Status	enum('0','1')	No	

Table 5.8: Structure of the table tbl_users

Column Name	Type	Null	Default
-------------	------	------	---------

<i>Id</i>	int(11)	No	
Book_id	int(11)	No	
User_id	int(11)	No	
Action	varchar(20)	No	
Creation_date	date	No	

Table 5.9: Structure of the table *tbl_book_preferences*

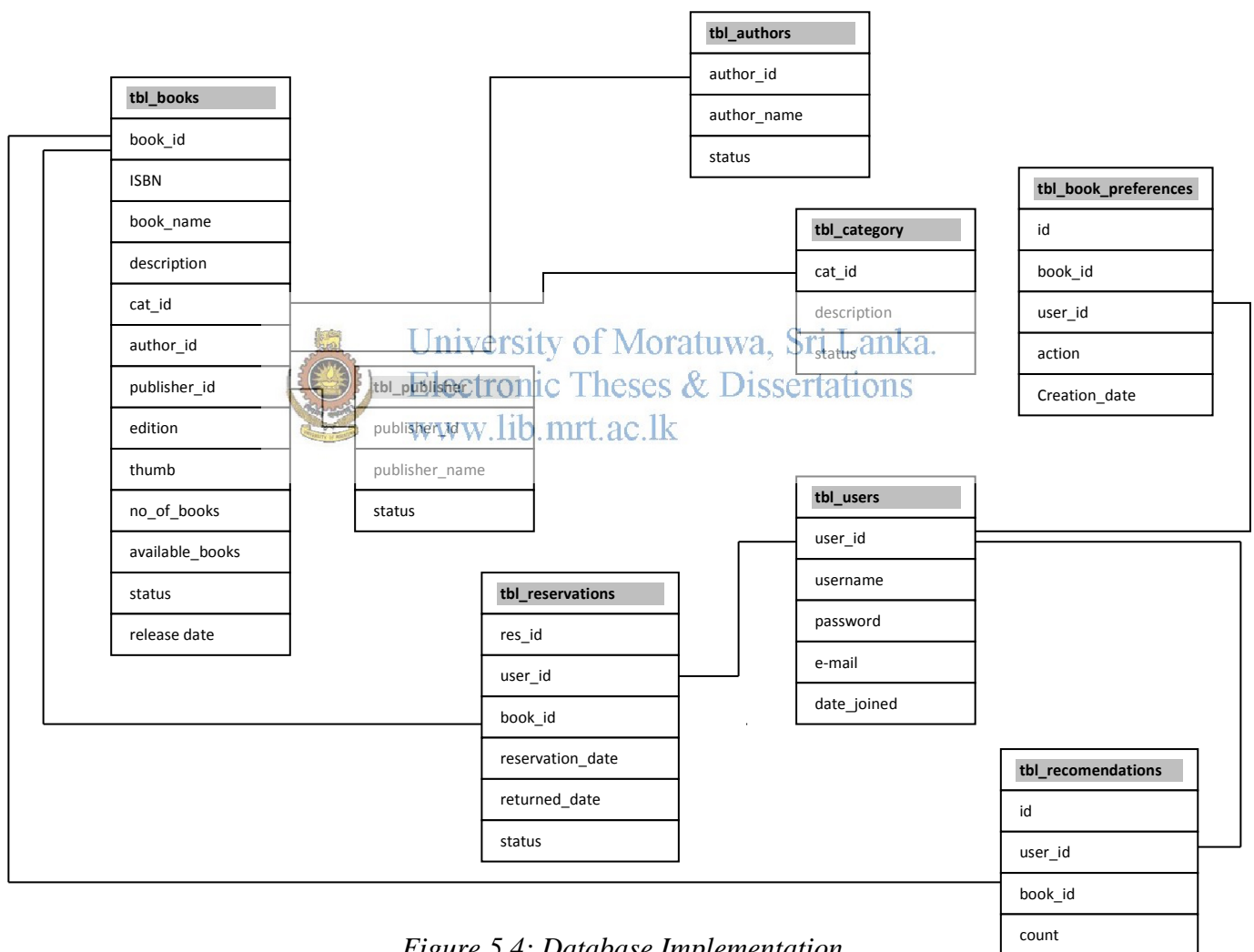


Figure 5.4: Database Implementation

5.6 Summary

The chapter has outlined the functional design and the database implementation of the application. The functionality of the system has been described using entity relationship diagram and use case diagrams.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Chapter 6

Implementation

6.1 Introduction

The book recommendation system has been implemented in line with the functional aspects outlined in the Chapter 5. The implementation has been made with the support of WAMP server simulation platform on a desktop environment. The system has the capability of functioning in a client server environment.

In order to access the system, Apache server and MySQL server must be made online. This could be done by activating the wampmanager.exe in \\wamp directory. The application could be called through the web browser by accessing the ‘elib’ program from the local host.

6.2 Recommendations on most viewed books

In this module system recommends books for the user by considering the most viewed books of the user.

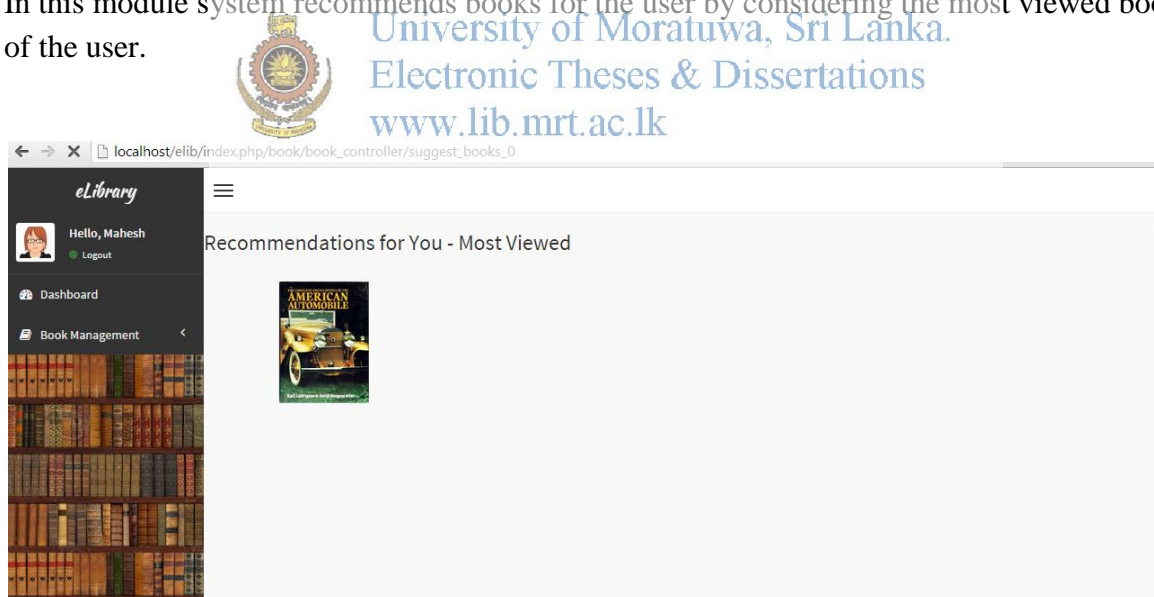


Figure 6.1: Recommendations based on most viewed books

6.3 Recommendations on reservation related books

In this module system recommends books for the user by considering the reservations of the other users

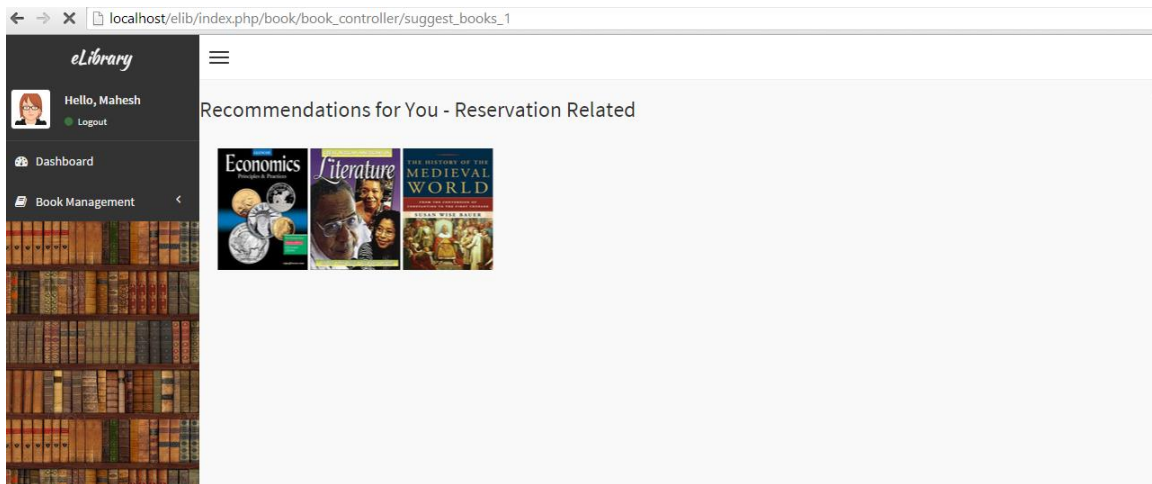


Figure 6.2: Recommendations based on reservation related books

6.4 Recommendations on most preferred books

In this module system recommends books for the user by considering the reservations of the other users



Figure 6.3: Recommendations based on most preferred books

6.5 Search Books

This allows the admin / librarian to search books by ISBN, book title, author, category or publisher.

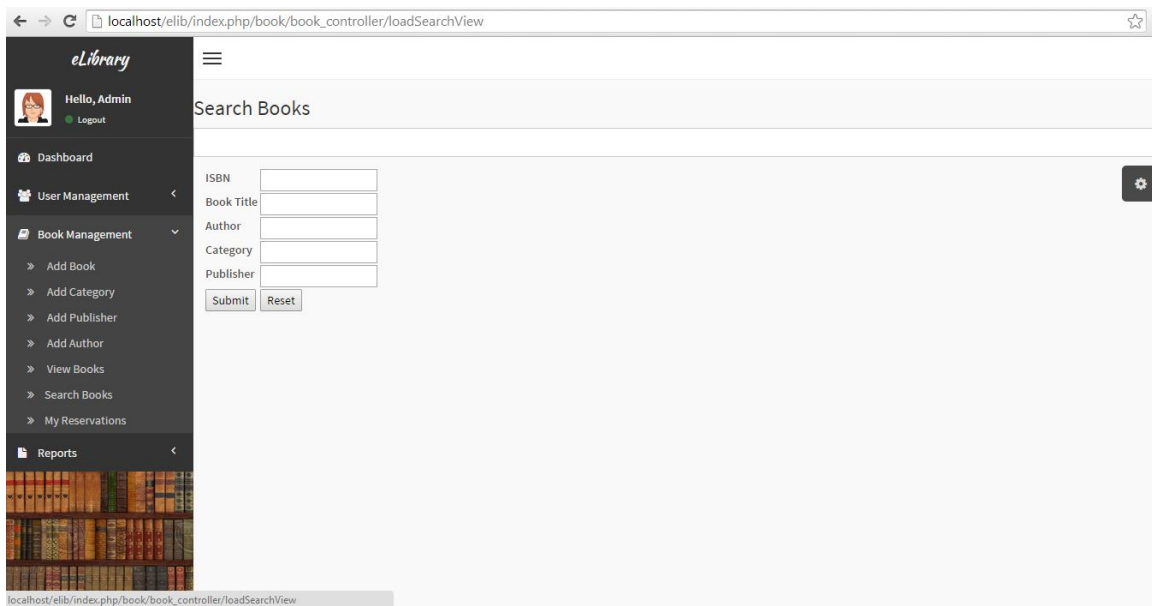


Figure 6.4: search books

6.6 View users

Allows admin user to view all the current members of the system.



Figure 6.5: View all the users of the system.

6.7 Reservation Report

Report which provides all the reservation details

Reservation ID	Book Title	Username	Reservation Date / Time	Due Date	Status
4	Economic Development	smith	2014/08/25 10:09:50	2014/09/08 10:09:50	Pending
5	Islamic Finance	smith	2014/08/25 10:13:22	2014/09/08 10:13:22	Pending
6	Economic Development	eric	2014/12/19 02:25:16	2015/01/02 02:25:16	Pending
9	Corporate Finance Law	maresh	2014/12/19 09:44:49	2015/01/02 09:44:49	Pending
10	Logic	smith	2015/01/04 09:19:13	2015/01/18 09:19:13	Pending
11	World of Science	peter	2015/01/04 09:30:03	2015/01/18 09:30:03	Pending
12	The Big Book	maresh	2015/01/06 07:06:55	2015/01/20 07:06:55	Pending
13	Essentials Of Logic	Nimal	2015/01/14 07:43:04	2015/01/28 07:43:04	Pending
14	Computers	john	2015/01/25 10:11:28	2015/02/08 10:11:28	Pending
15	Economic Development	john	2015/01/25 10:12:06	2015/02/08 10:12:06	Pending
16	English Litreature	john	2015/01/25 10:17:05	2015/02/08 10:17:05	Pending
17	World Litreature	peter	2015/01/25 10:17:56	2015/02/08 10:17:56	Pending
18	Logic	eric	2015/01/25 10:22:53	2015/02/08 10:22:53	Pending
19	Economic Development	eric	2015/01/25 10:36:27	2015/02/08 10:36:27	Pending

Figure 6.6: Reservation Report

6.8 All Books Report

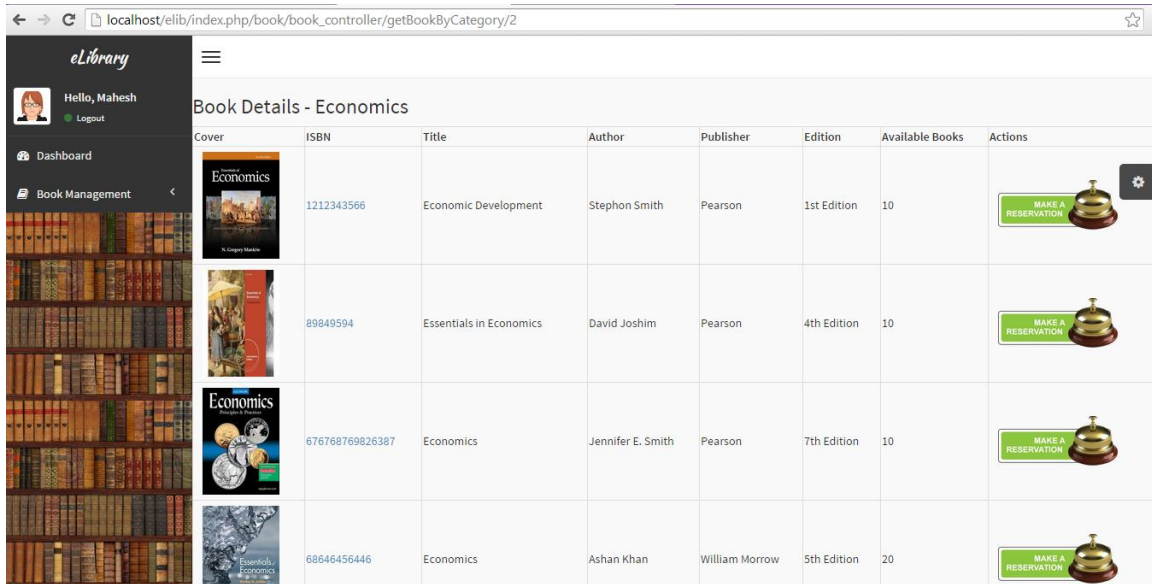
Report which provides details about all the books available in the library

Cover	ISBN	Title	Author	Publisher	Edition	Available Books	Actions
	1212343566	Economic Development	Stephon Smith	Pearson	1st Edition	8	MAKE A RESERVATION
	68646456446	Economics	Ashan Khan	William Morrow	5th Edition	20	MAKE A RESERVATION
	9098989898	Logic	Anne Gracie	Google	4th Edition	8	MAKE A RESERVATION
	7858483737	Digital Logic	Andrew Schloss	Usbourne	3rd Edition	10	MAKE A RESERVATION

Figure 6.7: All Books Report

6.9 Book reservation

Area of the system which provide facility to reserve books based on the category




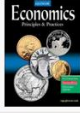

Cover	ISBN	Title	Author	Publisher	Edition	Available Books	Actions
	1212343566	Economic Development	Stephon Smith	Pearson	1st Edition	10	MAKE A RESERVATION
	89849594	Essentials in Economics	David Joshim	Pearson	4th Edition	10	MAKE A RESERVATION
	676768769826387	Economics	Jennifer E. Smith	Pearson	7th Edition	10	MAKE A RESERVATION
	68646456446	Economics	Ashan Khan	William Morrow	5th Edition	20	MAKE A RESERVATION

Figure 6.8: View book reservations

6.10 My Reservations

Area which allows user to find his current reservations



Cover	ISBN	Title	Reservation Date & Time	Due Date & Time
	67846324545	World of Science	2015-01-04 21:30:03	2015-01-18 21:30:03
	989384999	Nine Lives	2015-02-11 16:02:31	2015-02-25 16:02:31

Figure 6.9: My Reservations

6.11 Summary

The chapter has described the complete operating procedures of the application. It is important that all documentations must be completed at the end of the project. Key Operating Procedures must include all vital functions performed by all stakeholders.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Chapter 7

Evaluation

7.1 Introduction

The testing of the Recommendation System has been conducted under 4 main categories as given below.

1. Black Box Testing: This test has mainly been involved in testing the application-core. The test would be mainly aimed at the architecture of the application.
2. User Interface Testing: Testing user interfaces. This would be mainly focused on whether the user interfaces are working as expected, user friendliness, to ensure necessary feedback is provided as outputs at various stages, the relevance on the interfaces as per the functional requirements, and for the accuracy.
3. Functional Testing: Testing overall system requirements.
4. Acceptance Testing: User perspective of the system's quality.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

The test cases have been created in line with the requirement specifications and system design. The relevant test cases are described under below sections.

7.2 Black Box Testing

Following test cases have been defined in order to test the application core architecture. The tests are usually carried out by the high-level system owners of the application.

No.	Test Objective	Description	Result	Sign Off
	Whether the application could be installed as specified.	Perform installations of the application on Windows 7 and Windows XP with WAMP Server simulation software kit.		
	Check whether the	The Application and		

	application could be brought up as expected	database should be able to start up through the WAMP Console.		
	Whether the database and application interacts with each other	Check whether sample data keyed into the application would get stored in the tables of the database. Perform input, modification and deletion of data for all tables.		
	Whether the links between Knowledgebase database and other relevant databases are working fine	Check whether the application is able to link up with all relevant databases appropriately by interacting with databases at different stages.		
	Stability of the Security Module	Perform various tests with regard to the security of the database. Users should be able to perform activities pertaining only for the given level. Ensure other areas such as passwords, access and privilege levels are working as appropriately.		

Table 7.1: Black Box Testing

7.3 User Interface Testing

The is predominantly a application level testing in order to check whether all user interfaces of the client application are working fine. User interface is key to the stability of the environment as unstable interfaces might lead to garbage coming into the environment.

No.	Test Objective	Description	Result	Sign Off
	Admin User interface	Check creation of users. Check whether passwords work accordingly.		
	Normal User interface	Test all activities that need to be performed by the user. Check the menus, feedback when inputting and committing, error messages etc.		
	Appropriateness of Menus	Check whether the size of the menus are appropriate, language, accessibility.		
	Positioning of User interfaces	Whether the user interfaces are developed under best practices of Human Computer Interaction. Positioning of menus within the programs, whether the user is able to identify the options available?		

Table 7.2: Test Cases – User Interface Testing

7.4 Functional testing

The functional tests are carried out in order to check the entire system functionality. Generally this could be handled with the participation of administrators and end users. The functional testing would attract the highest user participation.

No.	Test Objective	Description	Result	Sign Off
	User Id creation	Whether the admin is able to create appropriate user classes. Whether such users granted the right level of privileges. Whether the passwords are working with the expected privileges etc.		
	Normal User functionality	Whether the user is able reserve books , search books		
	Recommendation of books	Whether user is able to get recommendations from the system to reserve books		


Table 7.3: test Cases – Functional Testing

7.5 Acceptance testing

Finally all stakeholders must go through all test results and jointly accept the system. The acceptance test could be performed jointly or respective teams individually.

7.6 Evaluate System

Finally quality of the software must be assessed in accordance with acceptable standards,

Criteria	Excellent	Good	Satisfactory	Weak
<p>Reliability</p> <p>Measure if product is reliable enough to sustain in any condition. Should give consistently correct results. Product reliability is measured in terms of working of project under different working environment and different conditions.</p>				
<p>Maintainability</p>  <p>Different versions of the product should be easy to maintain. For development its should be easy to add code to existing system, should be easy to upgrade for new features and new technologies time to time. Maintenance should be cost effective and easy. System be easy to maintain and correcting defects or making a change in the software.</p>				
<p>Usability</p> <p>This can be measured in terms of ease of use. Application should be user friendly. Should be easy to learn. Navigation should be simple.</p> <p>The system must be easy to use for input preparation, operation, and interpretation of output. Provide consistent user interface standards or conventions with our other frequently used systems. Easy for new or infrequent users to learn to use the system.</p>				

	<p>Portability</p> <p>This can be measured in terms of Costing issues related to porting, Technical issues related to porting, Behavioral issues related to porting.</p>				
	<p>Correctness</p> <p>Application should be correct in terms of its functionality, calculations used internally and the navigation should be correct. This means application should adhere to functional requirements.</p>				
	<p>Efficiency</p> <p>To Major system quality attribute. Measured in terms of time required to complete any task given to the system. For example system should utilize processor capacity, disk space and memory efficiently. If system is using all the available resources then user will get degraded performance. If system is not efficient then it cannot be used in real time applications.</p>				
	<p>Integrity or security</p> <p>Integrity comes with security. System integrity or security should be sufficient to prevent unauthorized access to system functions, preventing information loss, ensure that the software is protected from virus infection, and protecting the privacy of data entered into the system.</p>				
	<p>Testability</p> <p>System should be easy to test and find defects. If required should be easy to divide in different modules for testing.</p>				
	<p>Flexibility</p> <p>Should be flexible enough to modify. Adaptable to other</p>				

	products with which it needs interaction. Should be easy to interface with other standard 3rd party components.				
	<p>Reusability</p> <p>Software reuse is a good cost efficient and time saving development way. Different code libraries classes should be generic enough to use easily in different application modules. Dividing application into different modules so that modules can be reused across the application.</p>				
	<p>Interoperability</p> <p>Interoperability of one system to another should be easy for product to exchange data or services with other systems. Different system modules should work on different operating system platforms, different databases and protocols conditions.</p>				
	<p>Quality of interfaces</p>				



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
 Table 7.4: Quality Assurance Testing
www.lib.mrt.ac.lk

7.7 Usability Evaluation

Usability is a measure of interface quality that refers to the effectiveness, efficiency and satisfaction with which users can perform tasks. Evaluating usability is considered an essential part of the system development process.

Here user based usability evaluation is done by taking a sample set of users and allow them to use the system and fill the feedback form given below. Functionality of the system is categorized into 3 levels, while giving marks for each level as Easy -(10 marks) Moderate - (8 marks) Difficult - (5 marks)

User Level	Functionality	Easy	Moderate	Difficult
Sample admin users	Add new User	√		
	Add Book category	√		
	Add Author	√		

	Add book		√	
	Search Book by ISBN	√		
	Search Book by Category	√		
	Search Book by Author	√		
	Search Book by Release Date	√		
	View Books	√		
	View Reports	√		

Table 7.5: Usability Evaluation feedback for Admin users

based on the points in the above table the usability of the admin functions of the system is calculated $(9 \times 10) + 8 = 98$, therefore the usability of the admin functions is 98%.


User Level	Functionality	Easy	Moderate	Difficult
Sample Library members	Log in to system	√		
 University of Moratuwa, Sri Lanka. Electronic Theses & Dissertations reservebooks.ac.lk	View book categories	√		
	View each book	√		
	Reserve books	√		
	View recommendations	√		
	Reserve books	√		
	Search Book by ISBN	√		
	Search Book by Category	√		
	Search Book by Author	√		
	Search Book by Release Date	√		

Table 7.6: Usability Evaluation feedback for normal users

based on the points in the above table the usability of the functions of the system is calculated $(10 \times 10) = 100$, therefore the usability of the functions of the system is 100%.

By considering the above figures it is convinced that the usability of the system is high in terms of admin and normal user interactions.

7.8 Summary

Testing is vital for the smooth functionality of any application. Testing must be carried out with all stakeholders.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Chapter 8

Conclusion and Further Work

8.1 Introduction

The recommendation system has been developed in order to facilitate the stability assurance of the library functions of the organization. The entire system is developed based on MVC architecture. The most crucial part of the system is the recommendation engine which make automated suggestions for the members of the library when reserving books. The recommendation module should be improved by applying a collaborative filtering method which will increase the effectiveness of the recommendation mechanism of the system.

8.2 Achievement of the Aim and the Objectives of the project

The system has been able to demonstrate key functions of the library management, and the recommendation feature of the system is implemented based on content based filtering mechanism which basically focuses on the past behavior of the user and make suggestion to select books based on that.

8.3 Challenges of the project and Problems encountered

Main challenge faced in developing the recommendation system is the inability to implement the real time recommendations to the user.

8.4 Summary

The post implementation review must be conducted in order to obtain feedback from various parties. Such reviews often create the platform to make further improvements to the system. Also, would help for the development of systems of such nature in the future. Meanwhile, it is highly recommended that more emphasis is given to the enhancement of the recommendation engine in different angles through collaborative filtering of the application.



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

References

Chung, L. (n.d.). *2client.pdf*. Retrieved 02 15, 2015, from Website of University of Texas, Dallas: www.utdallas.edu/~chung/SA/2client.pdf

Francesco Ricci • Lior Rokach • Bracha Shapira, P. B. *Recommender Systems*. Springer US.

Gottwald, D. T. (2005). Recommender Systems for Libraries. *Recommender Systems for Libraries [Survey and Forecast]* , 70-75.

Group, T. P. (2001). *docs.php*. Retrieved 01 15, 2015, from Official PHP web site: <http://php.net/docs.php>

Jones, M. T. (2013, 12 12). *os-recommender1*. Retrieved 02 15, 2015, from IBM Official Website: <http://www.ibm.com/developerworks/library/os-recommender1/>

tutorialspoint. (n.d.). *basic_mvc_architecture.htm*. Retrieved February 15, 2015, from Tutorial Points Website: http://www.tutorialspoint.com/struts_2/basic_mvc_architecture.htm



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

Appendix A

Main Controller of the system

Book_controller.php is the base which includes all the major functions of the system.

```
<?php
```

```
if (!defined('BASEPATH'))
```

```
    exit('No direct script access allowed');
```

```
class Book_controller extends CI_Controller {
```

```
    public function index() {
```

```
        $this->viewCategories();
```

```
    }
```

```
    public function viewAllBooks() {
```

```
        if ($this->session->userdata('logged_in')) {
```

```
            $bk_model = new Book_model_service();
```

```
            $data['book_details'] = $bk_model->mGetAllBooks();
```

```
            $partials = array('content' => 'books/all_books'); //load the view
```

```
            $this->template->load('template/main_template', $partials, $data); //load the template
```

```
        } else {
```

```
            redirect('login/login_controller');
```

```
        }
```

```
    }
```

```
    public function viewCategories() {
```



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```

if ($this->session->userdata('logged_in')) {
    $bk_model = new Book_model_service();
    $data['book_cats'] = $bk_model->mGetAllCategories();
//    $data['recom_books']=$bk_model->mGetRecomsByUserHitCount();
    $partials = array('content' => 'books/categories'); //load the view
    $this->template->load('template/main_template', $partials, $data); //load the template
} else {
    redirect('login/login_controller');
}
}

public function getBookByCategory($cat_id) {
    if ($this->session->userdata('logged_in')) {
        $bk_model = new Book_model_service();
        $data['book_details'] = $bk_model->mGetBookByCategory($cat_id);
        $data['book_category'] = $bk_model->mGetCategoryById($cat_id);
        $partials = array('content' => 'books/all_books'); //load the view
        $this->template->load('template/main_template', $partials, $data); //load the template
    } else {
        redirect('login/login_controller');
    }
}

public function loadViewAddBook() {
    $book_model = new Book_model_service();
    $data['book_cats'] = $book_model->mGetBookCategories();
    $data['authors'] = $book_model->mGetAuthors();
    $data['publishers'] = $book_model->mGetPublishers();
    $partials = array('content' => 'books/add_book'); //load the view
    $this->template->load('template/main_template', $partials, $data); //load the template
}

```

```

public function loadViewAddCategory() {
    $partials = array('content' => 'books/add_category'); //load the view
    $this->template->load('template/main_template', $partials); //load the template
}

public function loadViewAddAuthor() {
    $partials = array('content' => 'books/add_author'); //load the view
    $this->template->load('template/main_template', $partials); //load the template
}

public function loadViewAddPublisher() {
    $partials = array('content' => 'books/add_publisher'); //load the view
    $this->template->load('template/main_template', $partials); //load the template
}

public function saveNewBook() {
    $book_model = new Book_model_service();
    $isbn = $this->input->post('isbn');
    $book_name = $this->input->post('book_name');
    $description = $this->input->post('description');
    $category = $this->input->post('category');
    $author = $this->input->post('author');
    $publisher = $this->input->post('publisher');
    $edition = $this->input->post('edition');
    $available_books=$this->input->post('available_books');
    $no_of_books = $this->input->post('no_of_books');
    $status = $this->input->post('status');
    $this->load->library('upload');
    $thumb = $_FILES['thumb']['name'];
    $path = base_url() . 'resources/books/';
    $allowedExts = array("gif", "jpeg", "jpg", "png");
    $temp = explode(".", $_FILES["thumb"]["name"]);

```

```

$extension = end($temp);

if ((($_FILES["thumb"]["type"] == "image/gif") || ($_FILES["thumb"]["type"] ==
"image/jpeg") || ($_FILES["thumb"]["type"] == "image/jpg") || ($_FILES["thumb"]["type"]
== "image/pjpeg") || ($_FILES["thumb"]["type"] == "image/x-png") ||
($_FILES["thumb"]["type"] == "image/png"))) && in_array($extension, $allowedExts)) {

    if ($_FILES["thumb"]["error"] > 0) {

        echo "Return Code: " . $_FILES["thumb"]["error"] . "<br>";

    } else {

        //echo "Upload: " . $_FILES["thumb"]["name"] . "<br>";

        //echo "Type: " . $_FILES["thumb"]["type"] . "<br>";

        // echo "Size: " . ($_FILES["thumb"]["size"] / 1024) . " kB<br>";

        // echo "Temp file: " . $_FILES["thumb"]["tmp_name"] . "<br>";

        if (file_exists("upload/" . $_FILES["thumb"]["name"])) {

            //echo $_FILES["thumb"]["name"] . " already exists. ";

        } else {

            move_uploaded_file($_FILES["thumb"]["tmp_name"], $path .
$_FILES["thumb"]["name"]);

            //die;

        }

    }

} else {

    echo "Invalid file";

    // die;

}

$save_status = $book_model->mSaveNewBook($isbn, $book_name, $description,
$category, $publisher, $author, $thumb, $edition, $no_of_books, $available_books, $status);

if ($save_status == 1) {

    redirect(site_url() . '/book/book_controller/viewCategories');

}

}

```

```

public function saveNewCategory() {
    $book_model = new Book_model_service();
    $category = $this->input->post('category');
    $status = $this->input->post('status');
    $save_status = $book_model->mSaveNewCategory($category, $status);
    if ($save_status == 1) {
        redirect(site_url() . '/book/book_controller/viewCategories');
    }
}

```

```

public function saveNewAuthor() {
    $book_model = new Book_model_service();
    $author = $this->input->post('author');
    $status = $this->input->post('status');
    $save_status = $book_model->mSaveNewAuthor($author, $status);
    if ($save_status == 1) {
        redirect(site_url() . '/book/book_controller/viewCategories');
    }
}

```

```

public function saveNewPublisher(){
    $book_model = new Book_model_service();
    $publisher = $this->input->post('publisher');
    $status = $this->input->post('status');
    $save_status = $book_model->mSaveNewPublisher($publisher, $status);
    if ($save_status == 1) {
        redirect(site_url() . '/book/book_controller/viewCategories');
    }
}

```

```

public function loadSearchView() {
    $partials = array('content' => 'books/search_books');
    $this->template->load('template/main_template', $partials);
}

```

```

public function searchBooks() {
    $book_model = new Book_model_service();
    $title = $this->input->post('title');
    $isbn = $this->input->post('isbn');
    $author = $this->input->post('author');
    $category = $this->input->post('category');
    $publisher = $this->input->post('publisher');
    $keyword = $this->input->post('keyword');
    $data['search_results'] = $book_model->mSearchBooks($title,
    $isbn,$author,$category,$publisher,$keyword);
    $partials = array('content' => 'books/search_results');
    $this->template->load('template/main_template', $partials, $data);
}

```



```

function loadReserveView($book_id){
    $book_model = new Book_model_service();
    $data['book_id']=$book_id;
    $data['book_details']=$book_model->mBookDetailsById($book_id);

```

```

$partials = array('content' => 'books/reserve_books');
$this->template->load('template/main_template', $partials, $data);
}

```

```

function reserveBook($bk_id,$avail_books){
    $book_model=new Book_model_service();

    $user_id=$this->session->userdata('user_id');
    $book_id=$bk_id;
    $available_books=$avail_books;
    $reserved_books=$book_model->mGetReservedBookCount($user_id);
    $reservation_status=$book_model->mCheckStatus($book_id,$user_id);

    if(count($reserved_books)<3){
        if($reservation_status){

            if($available_books>=1){
                $reserve_status=$book_model->mReserveBook($user_id,$book_id,$avail_books);
                if($reserve_status==1){
                    $data['book_details']=$book_model->mBookDetailsById($book_id);
                    $data['message']='Book Reserved Successfully !';
                    $partials = array('content' => 'books/reserve_books');
                    $this->template->load('template/main_template', $partials, $data);
                }
            }
            else{
                $data['book_details']=$book_model->mBookDetailsById($book_id);
                $data['message']='Error Occured !';
                $partials = array('content' => 'books/reserve_books');
                $this->template->load('template/main_template', $partials, $data);
            }
        }
    }
    else{

```



```

    $data['book_details']=$book_model->mBookDetailsById($book_id);
    $data['message']='Sorry, insufficient books for reservation !';
    $partials = array('content' => 'books/reserve_books');
$this->template->load('template/main_template', $partials, $data);
}

}

else{
    $data['book_details']=$book_model->mBookDetailsById($book_id);
    $data['message']='Sorry, you have already reserved this book !';
    $partials = array('content' => 'books/reserve_books');
$this->template->load('template/main_template', $partials, $data);

}

}

else{
    $data['book_details']=$book_model->mBookDetailsById($book_id);
    $data['message']='Sorry, you cannot reserve more than 3 books !';
    $partials = array('content' => 'books/reserve_books');
$this->template->load('template/main_template', $partials, $data)
}

}

public function loadReservationsView(){
    $book_model=new Book_model_service();

    $data['book_details']=$book_model->mReservationDetails();
    $partials = array('content' => 'books/reservation_list');
$this->template->load('template/main_template', $partials, $data);

```



University of Moratuwa, Sri Lanka.
 Electronic Theses & Dissertations
www.lib.mrt.ac.lk

```

}

public function bookRecommendations(){

    $book_model=new Book_model_service();

    $user_id=$this->session->userdata('user_id');

    $recommendations=$book_model->mGetRecommendations($user_id);

    return $recommendations;

}

public function getBookById($bk_id) {

    if ($this->session->userdata('logged_in')) {

        $bk_model = new Book_model_service();

        $data['book_details'] = $bk_model->mGetBookById($bk_id);

        $partials = array('content' => 'books/all_books');//load the view
        $this->template->load('template/main_template', $partials, $data); //load the template
    } else {

        redirect('login/login_controller');

    }

}

public function getPastReservations(){

    $book_model=new Book_model_service();

    $user_id=$this->session->userdata('user_id');

    $past_reservations=$book_model->mGetPastReservations($user_id);

    return $past_reservations;

}

}

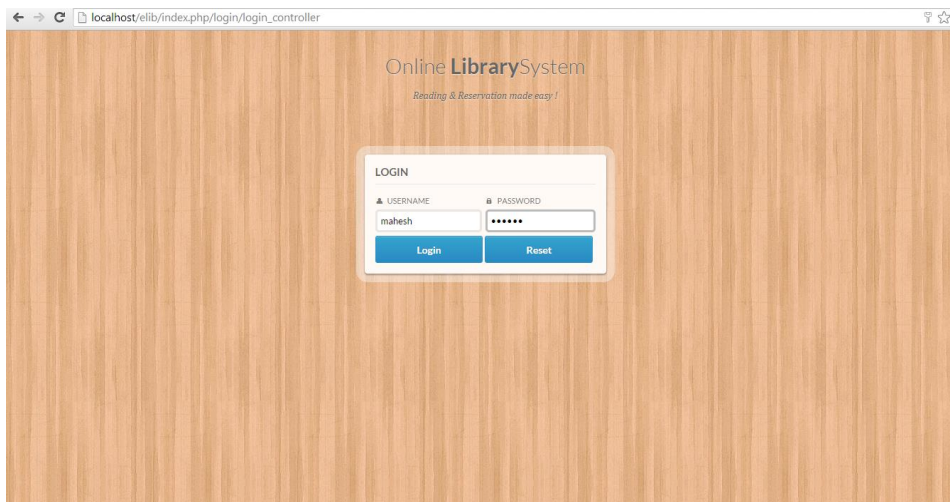
```



Appendix B

Screens of the System

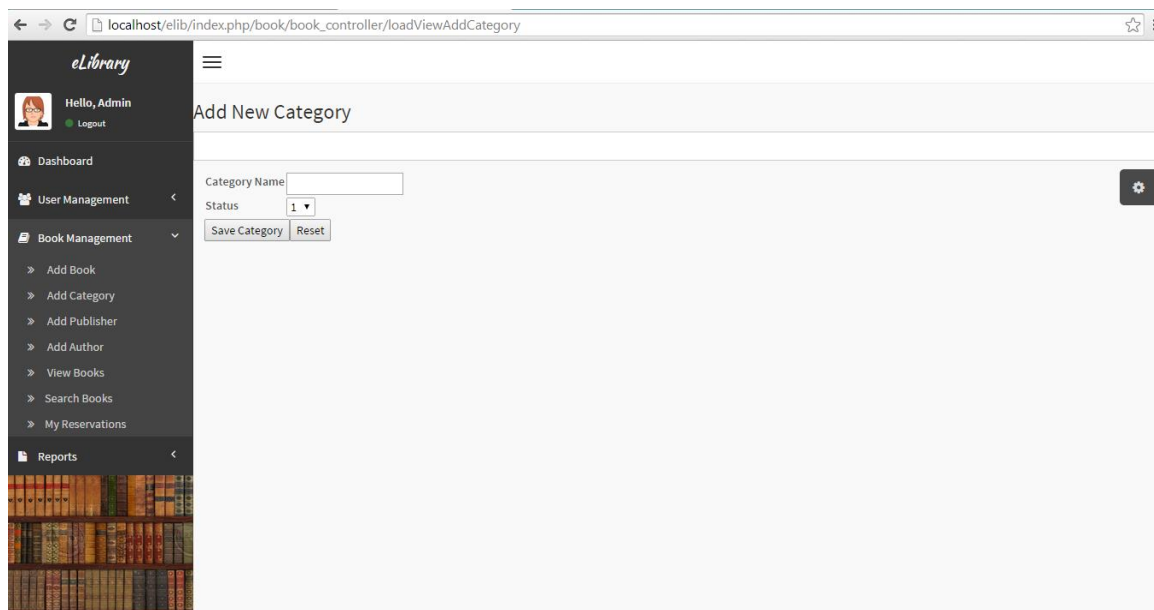
Login Screen



Admin Console



Add book category



The screenshot shows a web browser window with the URL `localhost/elib/index.php/book/book_controller/loadViewAddCategory`. The page title is "Add New Category". On the left, there is a sidebar menu for "eLibrary" with the user "Admin" logged in. The main content area contains a form with the following fields and controls:

- Category Name:** A text input field.
- Status:** A dropdown menu currently showing "1".
- Buttons:** "Save Category" and "Reset".

Add new publisher



The screenshot shows a web browser window with the URL `localhost/elib/index.php/book/book_controller/loadViewAddPublisher`. The page title is "Add New Publisher". On the left, there is a sidebar menu for "eLibrary" with the user "Admin" logged in. The main content area contains a form with the following fields and controls:

- Publisher Name:** A text input field containing "University of Moratuwa, Sri Lanka." and "Electronic Theses & Dissertations".
- Status:** A dropdown menu.
- Buttons:** "Save Publisher" and "Reset".

Below the form, there is a large watermark for the University of Moratuwa, Sri Lanka, with the text "Electronic Theses & Dissertations" and the website address "www.lib.mrt.ac.lk".

Add new author

The screenshot shows a web browser window with the URL `localhost/elib/index.php/book/book_controller/loadViewAddAuthor`. The page title is "Add New Author". On the left, there is a sidebar menu with the following items: Dashboard, User Management, Book Management (with sub-items: Add Book, Add Category, Add Publisher, Add Author, View Author, View Books, Search Books, My Reservations), and Reports. The main content area contains a form with the following fields: "Author Name" (text input), "Status" (dropdown menu with "1" selected), and two buttons: "Save Author" and "Reset". The top navigation bar includes the "eLibrary" logo, a user profile for "Admin" with a "Logout" link, and a hamburger menu icon. A decorative image of a bookshelf is visible at the bottom of the sidebar.

Add new user

The screenshot shows a web browser window with the URL `localhost/elib/index.php/user/user_controller/loadViewAddUser`. The page title is "Add New User". The top navigation bar features the "eLibrary" logo, a user profile for "Admin" with a "Logout" link, and a hamburger menu icon. A banner for the "University of Moratuwa, Sri Lanka. Electronic Theses & Dissertations" with the website `www.lib.mrt.ac.lk` is displayed. The left sidebar menu includes: Dashboard, User Management (with sub-items: Add User, View Users), Book Management, and Reports. The main content area contains a form with the following fields: "Username" (text input), "Password" (text input), "Re-enter Password" (text input), "Email" (text input), and "Status" (dropdown menu with "1" selected). There are "Save User" and "Reset" buttons at the bottom of the form. A decorative image of a bookshelf is visible at the bottom of the sidebar.

Library member area

localhost/elib/index.php/book/book_controller

Apps Google Maps Lesson 11: Passing f...

eLibrary

Hello, Mahesh
Logout

Dashboard

Home

Book Management

- View Books
- Search Books
- My Reservations
- Books Suggestions - Most Viewed
- Books Suggestions - Reservation Related
- Books Suggestions - Most Preferred

Book Categories

Category

- Science
- Economics
- Logic
- Literature
- Computer
- Mathematics
- Finance
- History
- Automobile



University of Moratuwa, Sri Lanka.
Electronic Theses & Dissertations
www.lib.mrt.ac.lk