

Library Automation System
For
Vidyarathna University College, Horana

B. R. DILRUKSHI

November 2017



Library Automation System
For
Vidyarathna University College, Horana

B. R. Dilrukshi

BIT Registration Number: R022201

Index Number: 0222011

Name of the Supervisor: Mr. Fayaz Thahir


2017 November



**This interim report is submitted in partial fulfillment of the requirement of the
Degree of Bachelor of Information Technology (External) of the
University of Colombo School of Computing**

DECLARATION


“I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the 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 abstract to be made available to outside organizations.”

Signature of Candidate : 

Date : 30-10-2017

Name of Candidate : B. R. Dilrukshi

Countersigned by:

Signature of Supervisor : 

Date : 30-10-2017

Name of Supervisor : Mr. Fayaz Thahir

ABSTRACT

With the changing modern society, people have to be automated to tally with the world. It is really worth to save time to move with the world and to get the work done easily. School library automation system project was chosen to apply that automation to the school library and to take more benefits out of it. The school which selected is, Vidyaratna University College, which is situated in Horana area. The school has a library which has approximately over 15,000 of library materials. The library mainly serves students, teachers and Vidyaratna Pirivena clergymen. With the increasing of the number of members and library materials, the usage of manual system for library may generate number of inefficiencies. Those are data duplication and gets more time to searching for particular item, recording of members, calculating overdue and circulating activities.

The proposed system facilitates to keep record of complete information of library materials. Automated library catalog is able to search based on any criteria which user has given. The system also provides easy way to make a check-out, check-in and hold library materials and facilitate to automatic fine calculation for late returns. The system mainly maintains user accounts and manages user activities. Finally facilitate to generate reports based on the criteria.

The system was developed using Hypertext Pre-processor (PHP) which is a server-side scripting language and Hypertext markup language (HTML), Cascading Style Sheets (CSS) which are client side scripting. Apache has been used as the web server; MySQL was used to handle databases. Additionally, Adobe Photoshop was chosen as the interface designing tool in the main stages of the developing. Rational Unified Process (RUP) was identified as the most suitable development methodology.

As this is a system with real time that is being expecting to overcome the issues, mentioned in the above and to cover all the necessary requirements. The system has clear instructions to guide user through the system and no special training is required. By validating user input data, accuracy has been achieved. Depending upon the category of user the access rights are decided. The system will meet user needs and fulfill the mission of the library and save the time of the reader.

ACKNOWLEDGEMENTS

First of all I would like to express my gratitude and thankful to the (Bachelor of Information Technology) BIT coordinator and academic staff of University of Colombo School of Computing (UCSC) for giving us valuable opportunity to follow this degree program and providing us a good guidance to successfully completion. Special thanks must go to my project supervisor Mr. Fayaz Thahir for his assistance and guidance to complete this project. I would like to thanks my client Vidyarathna University College, Horana. Ven. Opalle Ghanasiri Thero, principal of the school, Mr. S. Pallearachchi, deputy principal of the school, Mr. Laxman Kumara Aluthge, librarian of the school and the users of the library of the school for giving me great support and dedicated their valuable time to complete my project. In developing the system I am given many support and new ideas from my colleagues. It is my pleasure to thank all of them for their contribution to see my project a success.

TABLE OF CONTENTS

| | |
|--|-------------|
| DECLARATION..... | ii |
| ABSTRACT..... | iii |
| ACKNOWLEDGEMENTS | iv |
| TABLE OF CONTENTS | v |
| LIST OF FIGURES | viii |
| LIST OF TABLES | x |
| LIST OF ACRONYMS | xi |
| CHAPTER 1 – INTRODUCTION | 1 |
| 1.1 Introduction | 1 |
| 1.2 Motivation of the Project..... | 1 |
| 1.3 Scope of the Proposed Project..... | 2 |
| 1.4 Objectives of the System..... | 3 |
| 1.5 Structure of the Dissertation..... | 3 |
| CHAPTER 2 – ANALYSIS..... | 5 |
| 2.1 Introduction | 5 |
| 2.2 Analyzing the Current Manual System | 5 |
| 2.2.1 User Registration | 6 |
| 2.2.2 Circulation..... | 6 |
| 2.2.3 Card Catalogue..... | 6 |
| 2.2.4 Book Searching | 6 |
| 2.2.5 Calculate Fine | 6 |
| 2.2.6 Report Generation | 6 |
| 2.3 Similar Systems and Literature Review | 6 |
| 2.3.1 <i>Koha</i> Library Automation System | 7 |
| 2.3.2 Alice for Windows Library Automation System | 7 |
| 2.3.3 CDS/ISIS for Windows (Winisis) System..... | 8 |
| 2.4 Requirement Gathering | 8 |
| 2.4.1 Functional Requirements | 9 |
| 2.4.2 Non Functional Requirements | 11 |
| CHAPTER 03 – DESIGN..... | 13 |
| 3.1 Introduction | 13 |

| | | |
|---|---|-----------|
| 3.2 | System Development Life Cycle..... | 13 |
| 3.3 | Methodology for the Proposed System | 13 |
| 3.4 | Alternate Solutions | 14 |
| 3.4.1 | Standalone System | 15 |
| 3.4.2 | Software Collection | 15 |
| 3.4.3 | Mobile Based App | 15 |
| 3.4.4 | Reasons to Select the Web Based System | 16 |
| 3.5 | Object Oriented Designing | 16 |
| 3.5.1 | Use Case Diagram for the proposed School Library Automation System | 17 |
| 3.5.2 | Class Diagram for the proposed School Library Automation System.... | 18 |
| 3.5.3 | Entity Relationship Diagram for the proposed School Library Automation System..... | 18 |
| 3.5.4 | Sequence Diagram | 19 |
| 3.6 | User Interface Design for the System | 20 |
| 3.6.1 | Login Interface for the system | 21 |
| 3.6.2 | Home Page for the system | 22 |
| 3.6.3 | Data Entry Form | 22 |
| 3.6.4 | Data View Form..... | 23 |
| 3.6.5 | Member Search Form | 23 |
| CHAPTEY 04 – IMPLEMENTATION..... | | 25 |
| 4.1 | Introduction | 25 |
| 4.2 | Implementation Environment..... | 25 |
| 4.3 | System Developed Tools and Technologies | 26 |
| 4.4 | Code Features | 27 |
| 4.4.1 | Database Connection Page..... | 28 |
| 4.4.2 | Login Handling Page | 30 |
| 4.4.3 | Preventing the login details..... | 30 |
| 4.4.4 | Insert, Update and Delete Data | 31 |
| 4.4.5 | Data Validation | 33 |
| 4.5 | Re-Usable Components..... | 36 |
| CHAPTER 5 - EVALUATION..... | | 38 |
| 5.1 | Introduction | 38 |
| 5.2 | Software Testing Methods | 38 |

| | | |
|--|--------------------------------|-----------|
| 5.3 | Types of Testing | 39 |
| 5.3.1 | Unit Testing | 39 |
| 5.3.2 | Integration Testing | 39 |
| 5.3.3 | System Testing..... | 39 |
| 5.3.4 | Acceptance Testing | 39 |
| 5.3.5 | Regression Testing..... | 39 |
| 5.4 | Test Plan and Test Cases | 39 |
| 5.5 | User Evaluation | 45 |
| CHAPTER 06 – CONCLUSION | | 47 |
| 6.1 | Introduction | 47 |
| 6.2 | Future Improvements | 47 |
| 6.3 | Lessons Learnt..... | 48 |
| REFERENCES..... | | 49 |
| Appendix A – System Documentation..... | | 51 |
| Appendix B – Design Documentation | | 53 |
| Appendix C - User Documentation | | 61 |
| Appendix D – Management Reports..... | | 72 |
| Appendix E – Test Results | | 75 |
| Appendix F - Code Listing | | 77 |
| Appendix G - Client Certificate..... | | 88 |
| Glossary | | 89 |
| Index..... | | 91 |

LIST OF FIGURES

| | |
|--|----|
| Figure 2.1: Use case diagram for manual system | 5 |
| Figure 2.2: Koha Library Automation System - Home Page..... | 7 |
| Figure 2.3: Alice for Windows Library Automation System - Home Page..... | 7 |
| Figure 2.4: CDS/ISIS for Windows System - Home Page | 8 |
| Figure 3.1: Rational Unified Process Model..... | 14 |
| Figure 3.2: Use case diagram for the proposed system | 17 |
| Figure 3.3: Class diagram | 18 |
| Figure 3.4: Entity relationship diagram | 19 |
| Figure 3.5: Sequence diagram for user registration | 19 |
| Figure 3.6: Sequence diagram for check-out books..... | 20 |
| Figure 3.7: Sequence diagram for check-in books..... | 20 |
| Figure 3.8: Login interface | 21 |
| Figure 3.9: Administrator's home page | 22 |
| Figure 3.10: "New Member" data entry form..... | 22 |
| Figure 3.11: "User Information" data view form..... | 23 |
| Figure 3.12: "Member Search" form | 23 |
| Figure 3.13: Member information with details form | 24 |
| Figure 4.1: Three-tier architecture | 27 |
| Figure 4.2: Directory Structure | 28 |
| Figure 5.1: User evaluation questionnaire | 45 |
| Figure 5.2: Summarized result of user evaluation of the system | 46 |
| Figure B1: Member registration module..... | 53 |
| Figure B2: Item management module..... | 54 |
| Figure B3: Search items for check-out module | 55 |
| Figure B4: Check-out items module | 56 |
| Figure B5: Reserve an item module..... | 57 |
| Figure B6: Check-in item module..... | 58 |
| Figure B7: Generate reports module..... | 58 |
| Figure B8: Sequence diagram for member login..... | 59 |
| Figure B9: Sequence diagram for check-in with fine | 59 |
| Figure B10: Sequence diagram for item management..... | 60 |

| | |
|---|----|
| Figure C1: Login page of the system | 61 |
| Figure C2: Navigation panel and the dashboard | 62 |
| Figure C3: Logout button..... | 62 |
| Figure C4: Logout page | 62 |
| Figure C5: Library Catalog page | 63 |
| Figure C6: Circulation navigation | 63 |
| Figure C7: Member search form | 63 |
| Figure C8: Member information with equal name form..... | 64 |
| Figure C9: Member information with details form..... | 64 |
| Figure C10: New member form | 65 |
| Figure C11: Member information form | 65 |
| Figure C12: Exported data of user information | 66 |
| Figure C13: Check in form | 67 |
| Figure C14: Circulation navigation | 67 |
| Figure C15: Item search form | 68 |
| Figure C16: Administration navigation | 69 |
| Figure C17: Reports interface | 69 |
| Figure C18: Charts interface | 70 |
| Figure C19: Bar chart of login types..... | 70 |
| Figure C20: Pie chart of login types | 71 |
| Figure D1: Login type report | 72 |
| Figure D2: Member detail report | 73 |
| Figure D3: Check-out report | 74 |
| Figure G1: Client certificate | 88 |

LIST OF TABLES

| | |
|--|----|
| Table 4.1: Implementation details..... | 25 |
| Table 5.1: High level test plan | 42 |
| Table 5.2: User authentication | 42 |
| Table 5.3: Member registration module..... | 43 |
| Table 5.4: Check-out module..... | 44 |
| Table 5.5: Check-in module..... | 45 |
| Table A1: Hardware requirements..... | 51 |
| Table A2: Software requirements | 51 |
| Table B1: Use case description for member registration module..... | 53 |
| Table B2: Use case description for item management module..... | 54 |
| Table B3: Use case description for search items for check-out..... | 55 |
| Table B4: Use case description for check-out items..... | 56 |
| Table B5: Use case description for reserve an item..... | 57 |
| Table B6: Use case description for check-in item | 58 |
| Table B7: Use case description for report generation..... | 59 |
| Table E1: User authentication..... | 75 |
| Table E2: Member registration | 75 |
| Table E3: Check-out | 76 |
| Table E4: Check-in | 76 |

LIST OF ACRONYMS

| | | |
|--------|---|--|
| ABCD | – | Automation of liBraries and Centres of Documentation |
| BIT | – | Bachelor of Information Technology |
| CSS | – | Cascading Style Sheet |
| HTML | – | Hyper Text Mark-up Language |
| ILS | – | Integrated Library System |
| ISBN | – | International Standard Book Numbering |
| JS | – | Java Script |
| MARC | – | Machine Readable Catalogue |
| OB | – | OpenBiblio |
| PHP | – | Hypertext Pre-processor |
| RUP | – | Rational Unifies Process |
| SDLC | – | System Development Life Cycle |
| SQL | – | Structured Query Language |
| UCSC | – | University of Colombo School of Computing |
| UML | – | Unified Modeling Language |
| UNESCO | – | United Nations Educational, Scientific and Cultural Organization |

CHAPTER 1 – INTRODUCTION

1.1 Introduction

A school library is a library in a public or private elementary or secondary school that serves the information needs of its students and the curriculum needs of its teachers and staff, usually managed by a *school librarian* or media specialist. [1]

Beginning in the 1960s with the development of the Machine Readable Catalogue (MARC) record, the process of automation has expanded to include the core functions of acquisitions, cataloguing and authority control, serials control, circulation and inventory and inter library loan and document delivery. [2] Majority of the school libraries are using manual systems in Sri Lanka. When increasing of members and library materials, librarian was suffered; because there are lots of records to maintain. When there is a need to search for a particular item or member record, the librarian needs lot of time. Even a small mistake would create many problems. Security of information is very less.

A properly automated library will help its users with quick and prompt service. Developing a library automation system will help to increase the competency and efficiency of school library work, by reducing the heavy paper work. Further they can maintain and manage their day to day activities effectively and efficiently.

1.2 Motivation of the Project

Currently Library of Vidyarthna University College uses manual system. The library is maintaining heavy paper work and it is time consuming to retrieve data and also to handle day today library activities. The drawbacks of the existing manual system can be listed as follows,

- Time wasting paper works and highly labor-intensive to operate.
Particular searches on library materials or user information take longer and are less efficient.
- Using manual process to maintenance of library catalogue and member details.
- To identify issuing and returning dates of library resources gets more time.

- To find list of books borrowed by a particular member and calculate the overdue charges gets more time.
- Creating reports by referring manual records might be somewhat difficult.
- High tendency of Human Error
A manual library system relies heavily on the actions of people, which increases the possibility of human errors such as duplication of data entry.

To overcome the above mentioned issues, this library automation system was developed as the final year project. And this would be an opportunity to apply the gained knowledge to develop system software under the supervision of a real world component. The proposed library automation system will be facilitating to manage their activities more efficiently.

1.3 Scope of the Proposed Project

When deciding to develop a library automation system, the scope is most important evidence which we need to consider about. The scope of this project will be as follows:

- Provide facility to monitor and manage maintenance of library catalogue and arrangement of the books in the catalogue, maintenance of member details, issue dates and return dates, fine calculation efficiently.
- Searching facility to automated library catalog using title, author, subject, International Standard Book Numbering (ISBN), call number and publisher.
- Keep the data securely by providing relevant user authority to the system users.
- Implement barcode into the system. It is convenient and time saving as the user can direct scan the item's with the barcode id when the member borrows few items at one time.
- Alert member renewal, fine calculations and overdue items by email.
- Daily reports, weekly reports, monthly reports, quarterly reports and annual reports can be generated easily.

- Facilitate to the user to create an account and the user can access the system using user name and the password.

1.4 Objectives of the System

The main objective of this system is to provide a proper mechanism to manage the day to day school library activities which are done manually. Secondary objectives of the system are as follows:

- Reduce documentation to keep the records and supports to manage the time.
- Improve the efficiency of the book searching facility using automated library catalog.
- User management will secure the system data very well.
- By using barcode into the system, it will increase the efficiency of check-in and check-out process.
- Getting notification and relevant messages on time will increase the efficiency of the system.
- Generate reports to analyze the performances of the students and analyze the collection of the library on selected time periods.
- Provide an efficient system that any member can easily access.

1.5 Structure of the Dissertation

The dissertation contains six main chapters to provide an overall knowledge about the School Library Automation System. Dissertation structure as follows.

Chapter 02 - Analysis

The analysis chapter explains the difference between the manual system and the new automated system. Requirement gathering techniques that were used and the functional and nonfunctional requirement which are gathered, is mention in this chapter.

Chapter 03 - Design

The design chapter explains the methodical approach to the design of the solution and includes use case diagram, class diagram, entity relationship diagram, sequence diagram and main interfaces of the system.

Chapter 04 - Implementation

The implementation chapter explains the development procedure of the proposed system and includes hardware software requirements, development tools which are used at the time of development, code features and reused existing codes of the system.

Chapter 05 - Evaluation

Implementing and evaluation are depending with each other because testing is done in this phase. How the testing is done to make an accurate system is explained by this chapter using test cases.

Chapter 06 - Conclusion

The conclusion chapter explains the recommended future enhancements of the system and lessons learnt from the overall project work.

CHAPTER 2 – ANALYSIS

2.1 Introduction

System analysis is the study of a system and its components. System analysis will help to decompose a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose. Main objectives of this phase are identifying the system agents, their operations and relationships and data requirements. Before analyzing the system, first the requirements should be gathered by using the fact finding techniques, such as sampling of existing documentations, research and site visits, observations of the work environment, Questionnaires, interviews, prototyping and joint requirement planning. [3]

2.2 Analyzing the Current Manual System

Figure 2.1 shows the use case diagram of the current manual system.



Figure 2.1: Use case diagram for manual system

2.2.1 User Registration

Currently the school library has no systematic way to register users. Librarian stores details about users manually. In manual system they store student registration number, name, address, grade and contact number.

2.2.2 Circulation

The processes of check-in and check-out items were carried out manually.

2.2.3 Card Catalogue

Holdings of a library are printed, typed or handwritten on catalog cards. Each catalog card represents a single bibliographic item in the collection. Catalog cards are normally filed in a single alphabetical sequence or in separate sections by author, title and subject in the long narrow drawers of a specially designed filing cabinet usually made of wood or steel.

2.2.4 Book Searching

Members search books by using card catalogue.

2.2.5 Calculate Fine

Librarians calculate fines manually.

2.2.6 Report Generation

Librarians generate reports manually.

2.3 Similar Systems and Literature Review

When considering the School Library Automation System, there are open source and proprietary software. Most of the systems are very complex and not purely matching school libraries of our country. By studying current library automation systems, more experiences could be obtained about how the developing system should be and how the required functionalities should be presented. Following are a few similar systems that were reviewed to build the system.

2.3.1 Koha Library Automation System

Koha is an open source Integrated Library System (ILS), used world-wide by public, school and special libraries. In use world-wide, its development is steered by a growing community of users collaborating to achieve their technology goals. *Koha*'s feature set continues to evolve and expand to meet the needs of its user base. [4]

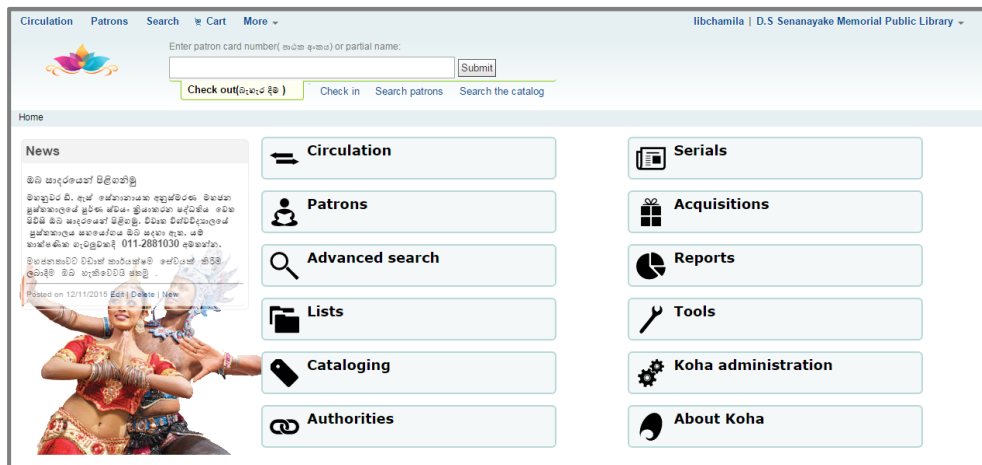


Figure 2.2: Koha Library Automation System - Home Page

2.3.2 Alice for Windows Library Automation System

Alice for Windows, an integrated library automation software package is the proprietary product of Softlink used by public library, school library, academic and further education libraries, special libraries, legal libraries, health and medical libraries. The software currently supports more than 10,000 organizations across 108 nations. [5]

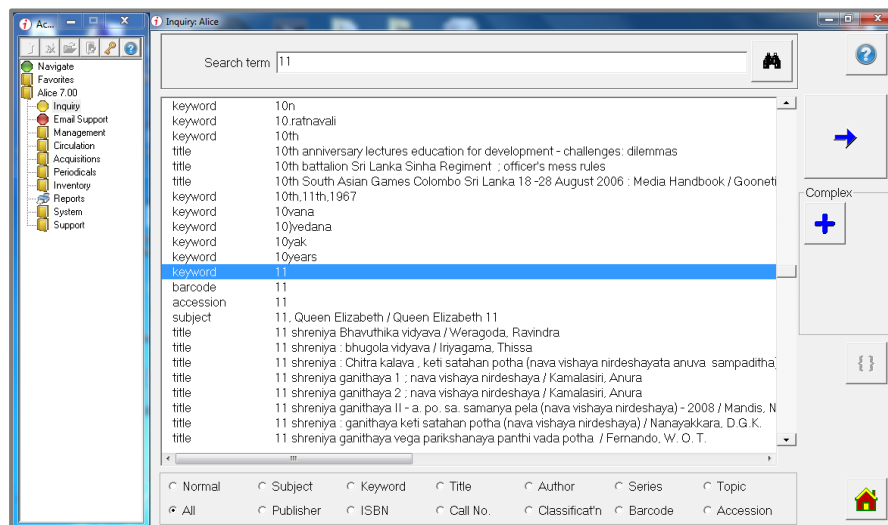


Figure 2.3: Alice for Windows Library Automation System - Home Page

2.3.3 CDS/ISIS for Windows (Winisis) System

CDS/ISIS is a software package for generalized Information Storage and Retrieval systems developed, maintained and disseminated by United Nations Educational, Scientific and Cultural Organization (UNESCO). It was first released in 1985 and since then over 20,000 licenses has been issued by UNESCO and a worldwide network of distributors. It is particularly suited to bibliographical applications and is used for the catalogues of many small and medium-sized libraries. [6]

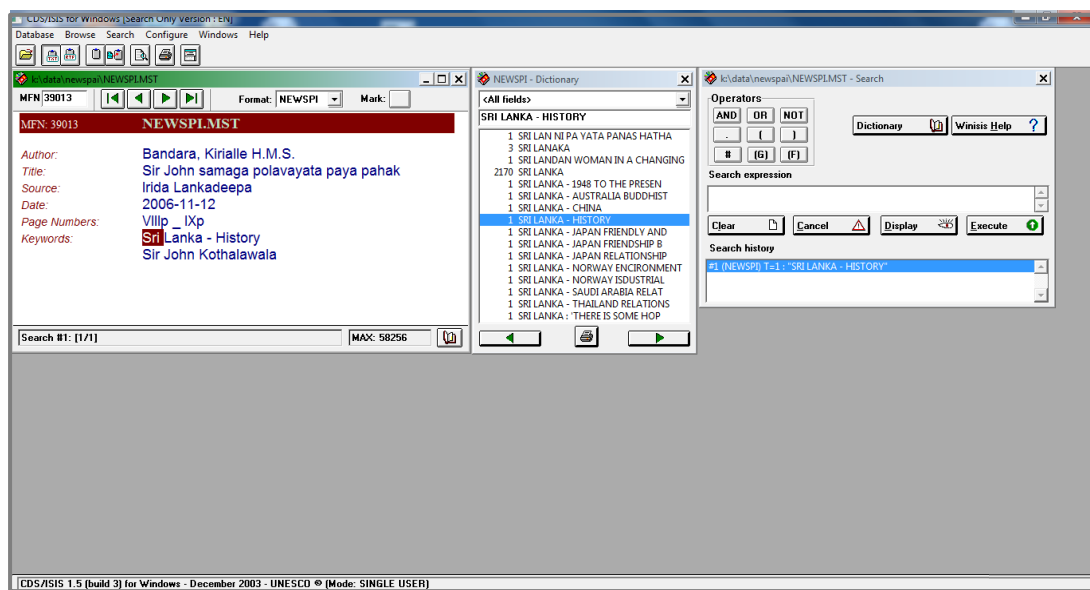


Figure 2.4: CDS/ISIS for Windows System - Home Page

2.4 Requirement Gathering

The analyst needs to collect facts and all relevant information to study any system. Collecting required facts are very important to apply tools in system development life cycle. The success of any project is depending upon the accuracy of available data. Accurate information can be collected with the help of certain techniques. These specific methods for finding information of the system are termed as fact finding techniques. The analyst may use more than one technique for investigation. There are several fact finding techniques which can be used to collect the clear and accurate information. In this project, facts were gathered by using following techniques.

- Sampling of existing documentations
- Observations of the work environment

- Questionnaires
- Interviews

The best way to analyze the existing system is to collect facts from existing documentation rather than from human sources. User complaints, suggestion box notes, reports, e-mails, various types of flowcharts and diagrams, program documentation and user training manuals are some kind of documents to collect facts from existing system.

Collecting facts can be carried out either participates in or watches a person perform activities to learn about the system. Normally system analyst observes techniques such as study the flow of documents in the existing system and interacts with the users. Observations can be a useful technique when the system have user point of view.

Interview is the most commonly used technique to collect information from individuals through face-to-face interaction. Purpose of an interview is to find, verify, clarify facts, motivate end users involvement, identify requirements and gather ideas and opinions. Interviews allow the system analyst to probe for more feedback from the interviewee. It becomes greater advantage to this project.

Requirements for the proposed system were gathered mainly by interviewing the targeted user group (Librarian, Staff, Teacher and Student) of the system. The background knowledge required is gathered by observing the existing automated systems and manual systems. The gathered requirements will be discussed below of this chapter.

2.4.1 Functional Requirements

What the system should do, how the system should react to particular inputs, and how the system should behave in particular situation is known as functional requirements. These requirements depend on the type of software being developed, the expected users of the software, and the general approach taken by the organization when writing requirements. [7]

Catalogue Management

Facilitate to create machine readable bibliographic record for a computerized catalog.

Circulation Management

The system allows check-in and check-out library materials and calculating the total number of items checked-out by library borrowers over a designated period of time and to the number of times a given item is checked-out during a fixed period of time. The system facilitate to identify weeding out items from the collection which is low circulation criterion and identifying ordering multiple copies which is high circulation criterion.

Barcode Scanning

Allows the user to accurately identify books and other materials for circulation, inventory and to link the borrower's library card to the appropriate member record.

Search Functionality

The system allows the user to search the automated library catalog using author, title, subject and keywords and allow users to print, download or export records. It displays bibliographic data of a book such as author, title, subject, edition, year of publication and publisher.

Fine Calculation

Facilitate to calculate fines and warning every ten days via email message.

Report and Charts Generation

The system has a report feature that will allow the user to generate a report showing the information about the all users who have overdue books and penalty, shows the information of all the check-out items in a time period which is the search criteria input by user, shows the information about all acquired items, shows the information about all registered members and shows the statistical information about all the library items categorized by item type or subject and statistical information about members

categorized by member type. The system shall generate those reports to the display a file or a print that is linked to the system.

User Management

The user with valid user name and password can access the Library Automation System. The system will give the response for invalid user name and password. The system displays the user account information including user id and password. The system allows administrator and librarian to select actions including removing, editing, adding user account and account information.

2.4.2 Non Functional Requirements

Non-functional requirements are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services. [7]

Security

The system should provide database modification only for librarians and system administrator after authorization procedures and access to the system is permitted only for library staff, teachers and college student after authorization procedures.

Usability

The interface should be designed with user friendly and easy to use by the staff so that the user can perform their job nicely. It should have a clear instruction to guide user through the system. The product should be used by people without training.

Accuracy

Accuracy is also an important nonfunctional requirement which we need to consider. If we do not consider about the accuracy, database issues can be occurred. By applying validation techniques to the user input data, accuracy has been achieved in this system.

Performance

Any interface between a user and system should have a maximum response time of 5 seconds. The system should be available for use 24 hours per day, 365 days per year.

Efficiency

Librarian should be able to process faster when they process book transactions. With the use of barcode scanner, librarian can avoid typing the book id one by one, barcode scanner enables librarian to scan the book id instantly.

CHAPTER 03 – DESIGN

3.1 Introduction

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. [8]

3.2 System Development Life Cycle

The system development life cycle is a process includes all of the activities involved in the system. It consists of requirement definition, system design, hardware and software engineering, system integration and testing. The objective of the System Development Life Cycle (SDLC) is producing the high quality software. There are various SDLC models such as waterfall model, prototyping model, evolutionary model, incremental development model, rational unified process model and agile process model.

3.3 Methodology for the Proposed System

Rational Unified process (RUP) has been selected for the proposed system. RUP is a modern process model that has been driven from work on the Unified Modeling Language (UML) and the associated Unified Software Development Process. RUP is normally described a dynamic perspective (which shows the phases of the model over time), a static perspective (which shows the process activities that are enacted) and a practice perspective (which suggests good practices to be used during the process). School Library automation system has been divided into models such as Catalogue Management and Circulation Management. Each model has developed incrementally and iteratively. Inception, Elaboration, Construction and Transition are the main phases of the RUP. [7]

Inception Phase

Inception phase, identify all external entities that will interact with the system and define the interactions and finally establish a business case for the system.

Elaboration Phase

Elaboration phase, develop the project plan and identify key project risks and get a clear idea about the problem domain.

Construction Phase

Construction phase, involves system design, programming and testing.

Transition Phase

Transition phase, system has been moved from the developers to the end users.

The following Figure 3.1 illustrates the phases of the RUP life cycle.

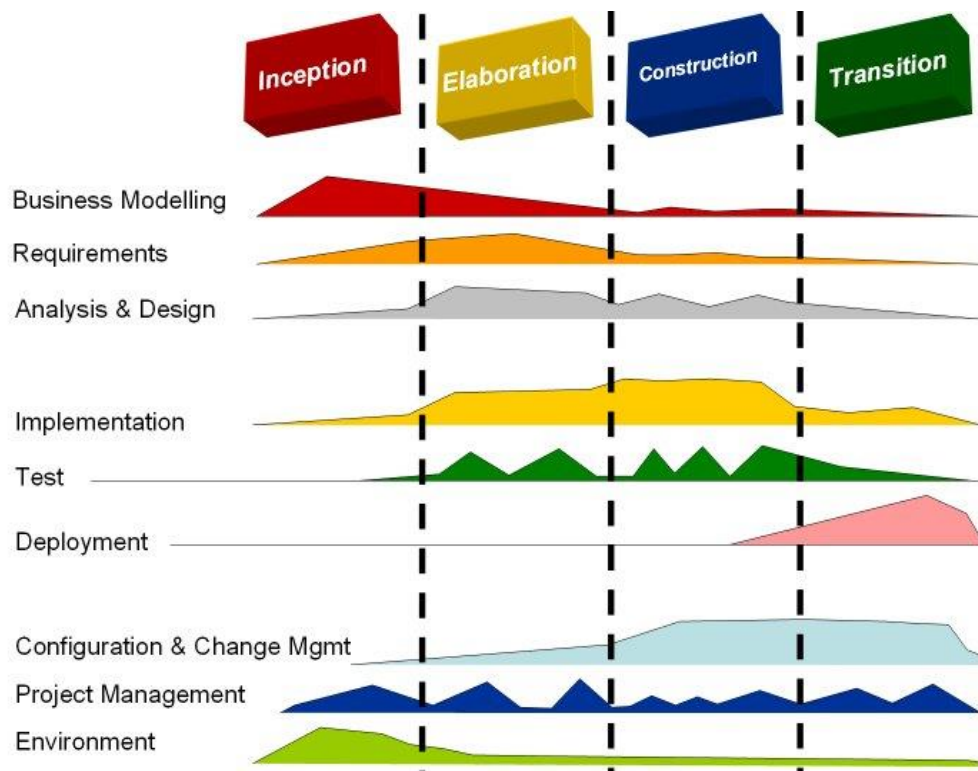


Figure 3.1: Rational Unified Process Model

3.4 Alternate Solutions

The alternate solutions for the web based school library automation system are; developing a standalone system for the library automation, having a collection of software to fulfill the library activities, having a mobile based app to automate library activities or cling to the traditional manual library system.

3.4.1 Standalone System

Standalone system is a software program that is not part of some bundled software and that does not load any external module more than the operating system to run. Those systems are run as a separate computer process and every terminal must be maintained individually. Most of the standalone systems are platform dependent. Setting up the particular standalone operating environment could be very expensive. The deployment, updating and maintenance processes are time consuming.

3.4.2 Software Collection

To fulfill the library automation process using a collection of software could be achieved by acquiring;

- Library catalog management software to maintain check-out, check-in and hold activities.
- Order management software to maintain acquisition activities.
- Report generation software to generate reports.

Maintain a collection of software for separate tasks might be expensive and the library might have to pay for the features that the library does not even use.

3.4.3 Mobile Based App

A mobile app is a type of computer program designed to run on a mobile device such, as a smartphone or tablet computer.

Mobile apps make your library system easily accessible to members. But it takes more time and money to develop. Library system maintains lot of data, when increasing the capacity of data, mobile device may be overload and the system will be slow. With the limited hardware resources, system may face various barriers.

3.4.4 Reasons to Select the Web Based System

- Cost effective

Web based applications can be used from almost any device with a browser. There is no need to develop and test it on all possible operating system versions and configurations.

- System would be platform independent.

Web Based System requires an up to date browser such as Internet Explorer, Google Chrome or Mozilla Firefox. Whether the user runs Windows, Mac OS or Linux, the system is built for the browser.

- Easier installation and maintenance.

With the web-based approach, installation and maintenance becomes less complicated. It needs only be installed or upgraded on the server. All the users can access it straight away and there is no need to upgrade the PC of each and every potential user.

- Easy and quick development cycle.

In general, though, web based languages like HTML, JS, and CSS are easier to code-in and provide quicker results. There are many frameworks in place to make the job even easier.

3.5 Object Oriented Designing

Object oriented design processes involve designing object classes and the relationships between the classes. The classes define the objects in the system and their interactions. When the design is realized as an executing program, the objects are created dynamically from these class definitions. Object oriented systems are easier to change than systems developed using functional approaches. Changing the implementation of an object or adding services should not affect other system objects. [7]

3.5.1 Use Case Diagram for the proposed School Library Automation System

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. The "actors" are people or entities operating under defined roles within the system. [9]

Use Case diagram is powerful concept for helping an analyst to understand how a system should behave. It helps to gather requirements from the users' point of view. Figure 3.2 shows the use case diagram of the proposed school library automation system.

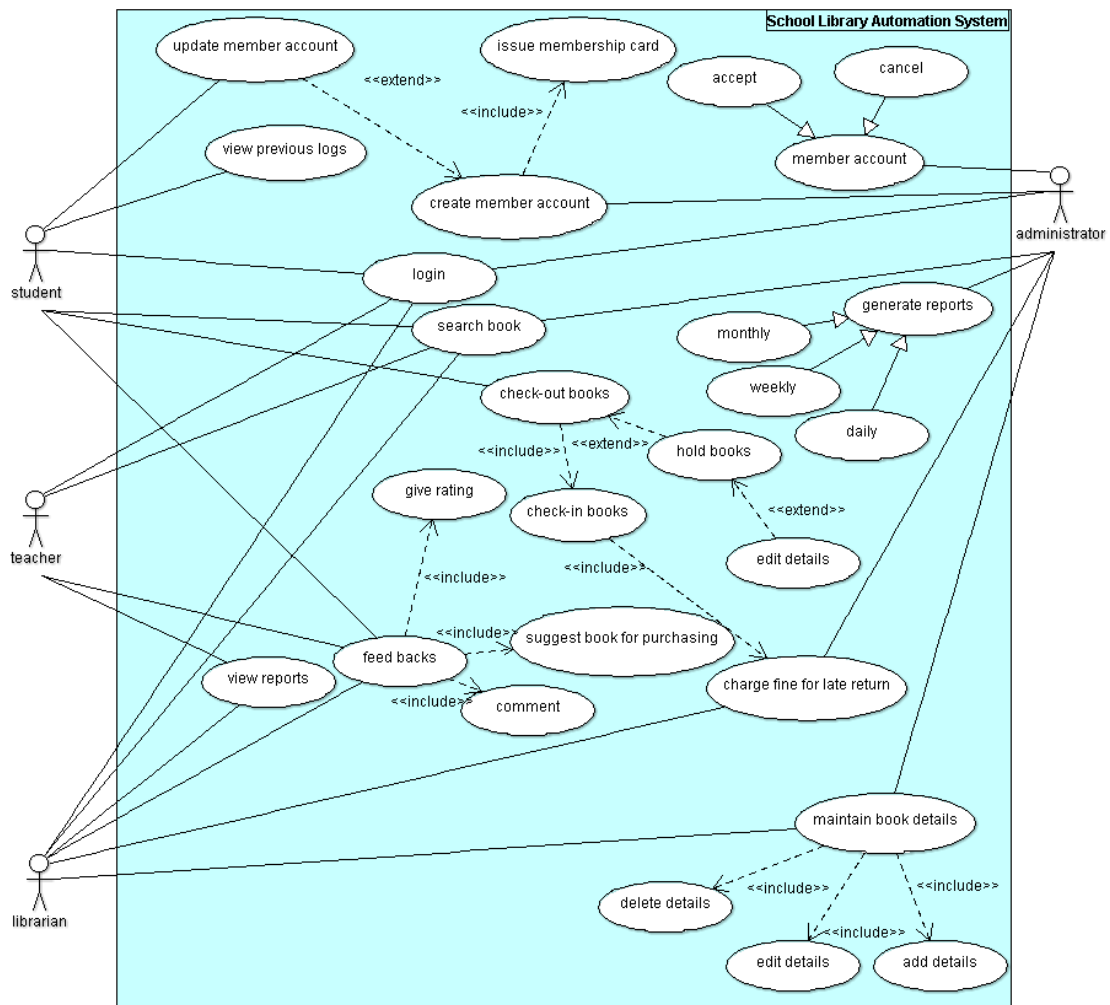


Figure 3.2: Use case diagram for the proposed system

3.5.2 Class Diagram for the proposed School Library Automation System

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. [10]

Figure 3.3 shows the class diagram of the proposed school library automation system.

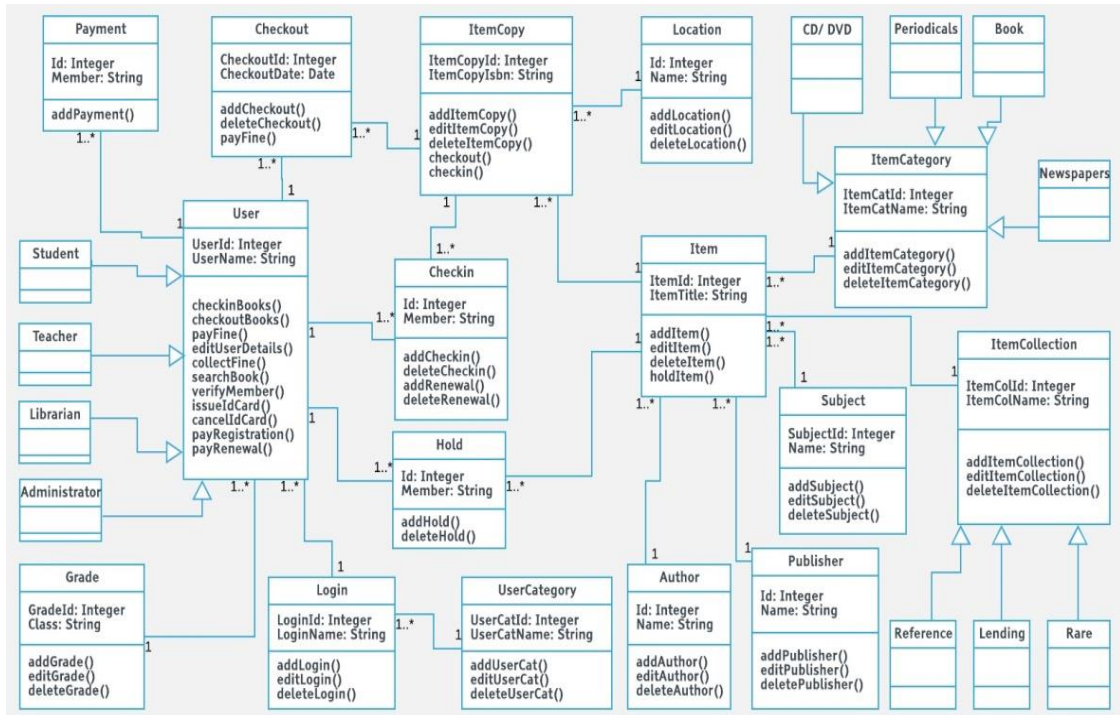


Figure 3.3: Class diagram

3.5.3 Entity Relationship Diagram for the proposed School Library Automation System

Entity Relationship diagrams shows the dynamic behavior of a system.

Figure 3.4 shows the Entity Relationship diagram of the system

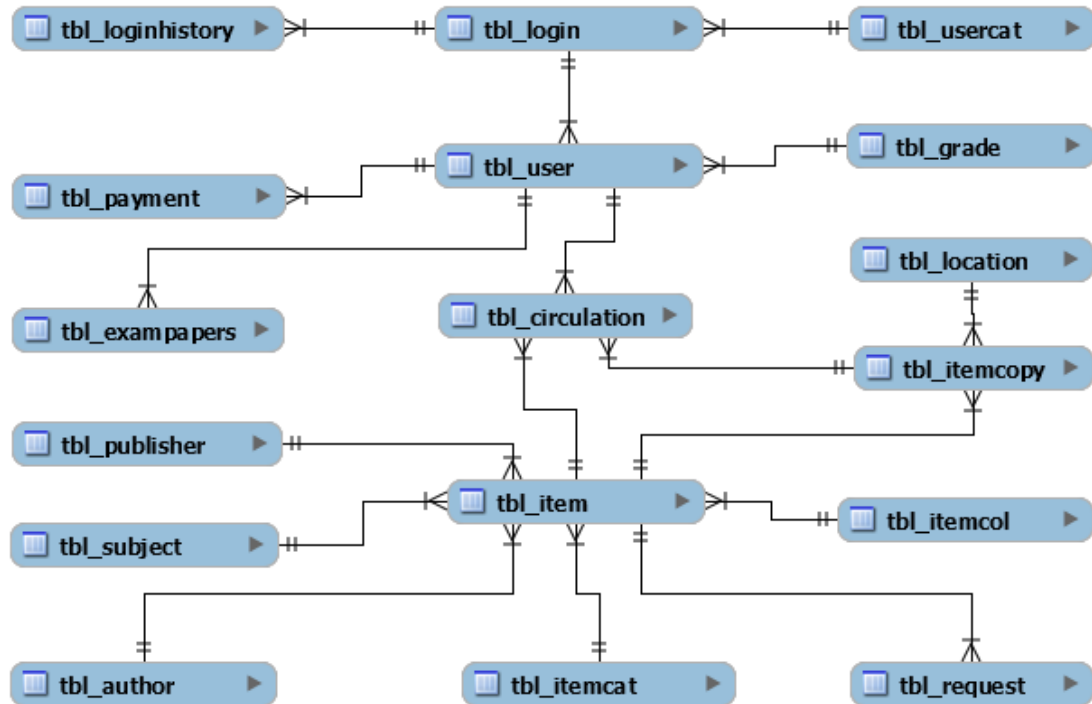


Figure 3.4: Entity relationship diagram

3.5.4 Sequence Diagram

Sequence diagram show the time-based dynamics of the interaction of a system. Figure 3.5 shows the sequence diagram for user registration.

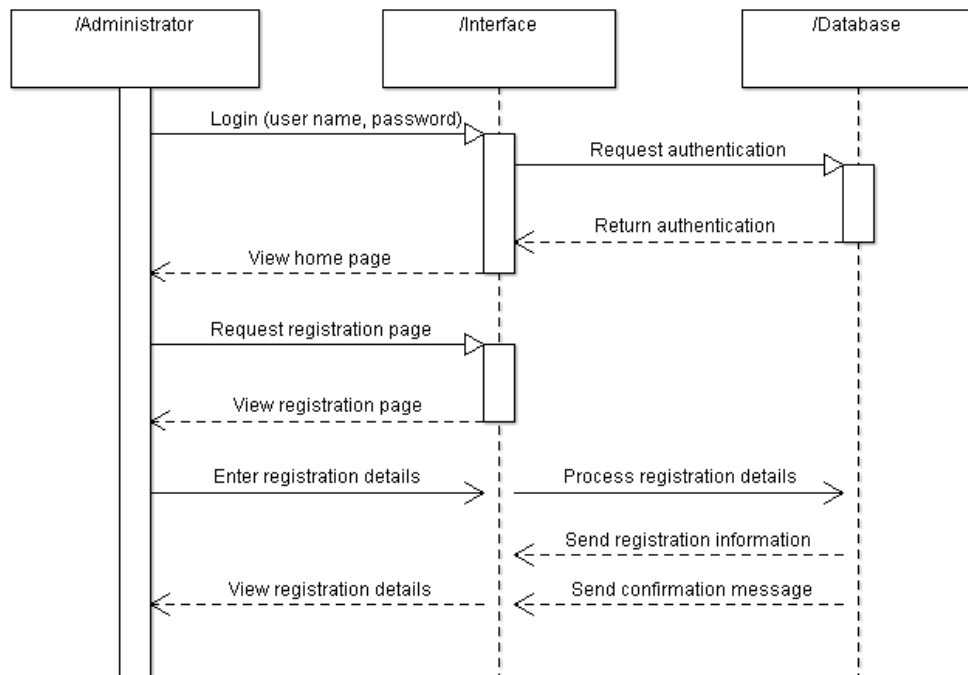


Figure 3.5: Sequence diagram for user registration

Figure 3.6 shows the sequence diagram for check-out books

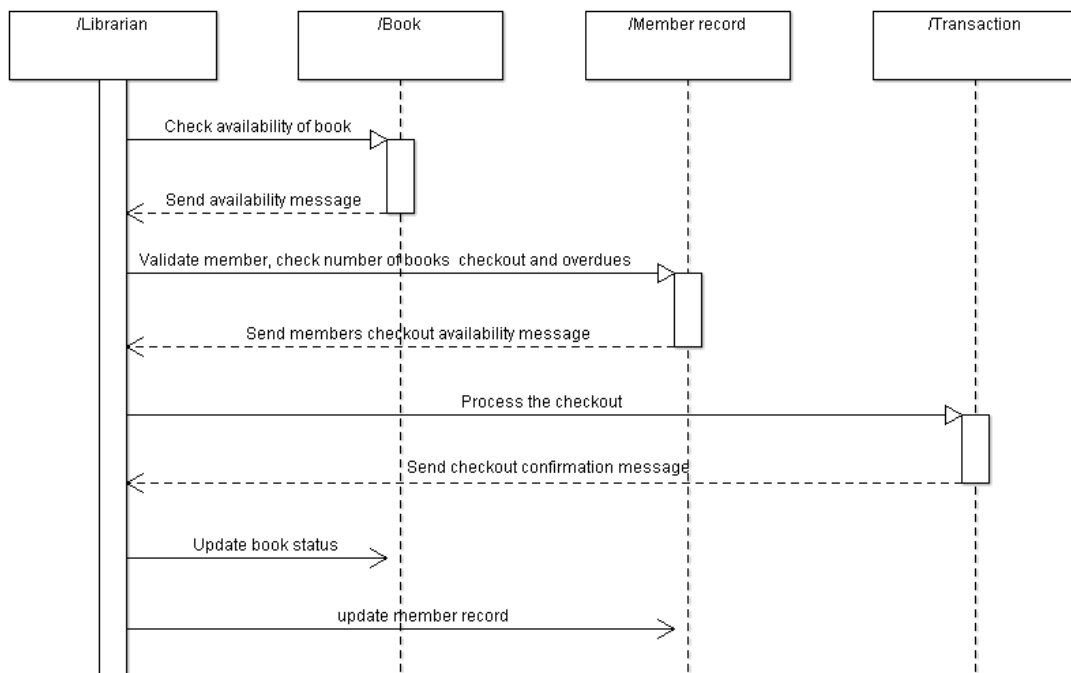


Figure 3.6: Sequence diagram for check-out books

Figure 3.7 shows the sequence diagram for check-in books

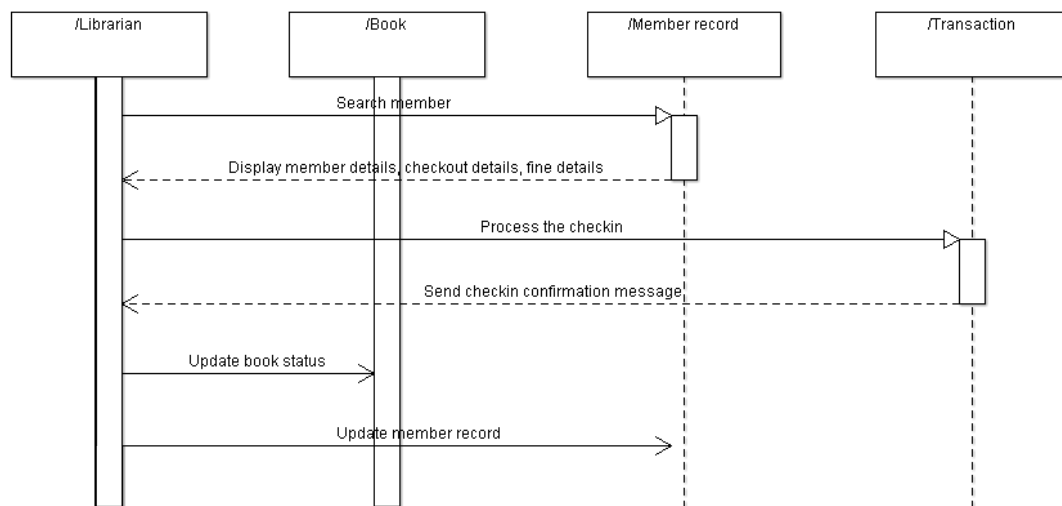


Figure 3.7: Sequence diagram for check-in books

3.6 User Interface Design for the System

Interface design is critical to the success of a system. An interface that is difficult to use, will results a lot of user errors. At worst, user will simply refuse to use the

software system irrespective of its functionality. If information is presented in a confusing or misleading way, users may misunderstand the meaning of information. They may initiate a sequence of actions that corrupt data or even cause catastrophic system failure. The system should assist the user providing help facilities and should guide the user in the case of occurrence an error. User interface design principles are as follows. [7]

User familiarity - The interface should use terms and concepts which are drawn from the experience of the people who will make most use of the system.

Consistency - The interface should be consistent in that, where ever possible, comparable operations should be activated in the same way.

Recoverability - Include mechanisms to allow users to recover from errors.

User guidance - Provide meaningful feedback when errors occur and provide context-sensitive user help facilities.

User diversity - The interface should provide appropriate interaction facilities for different type of system user.

3.6.1 Login Interface for the system

This is a common interface for all users to provide the facility to login to the system. Only authorized users can access into the system. If the invalid user log into the system, error messages are popup.

Following figure 3.8 shows login interface of the system.

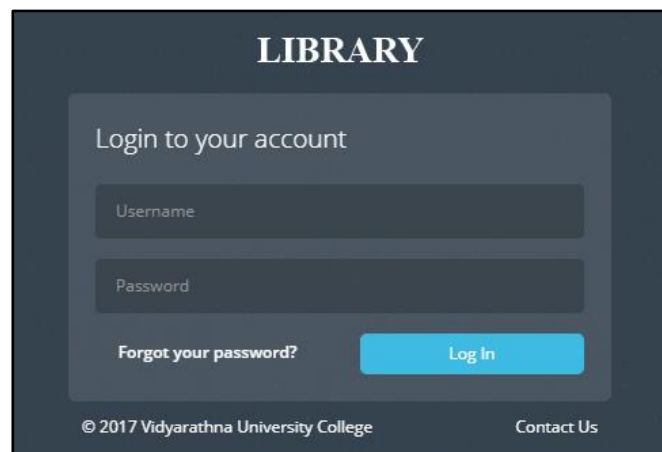


Figure 3.8: Login interface

3.6.2 Home Page for the system

The home page for the system will be displayed after the successful login. It varies according to the user group and provides dash board with module access, navigation bar and few links for quick information to let the user to navigate easily through the system. Figure 3.9 shows an administrator's home page.

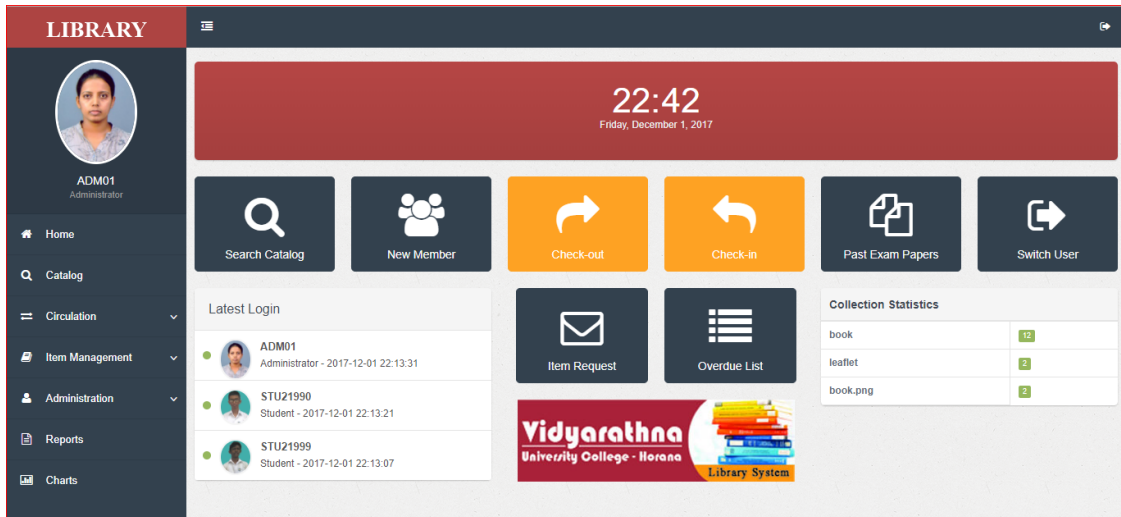


Figure 3.9: Administrator's home page

3.6.3 Data Entry Form

The system has data entry/ edits forms for each module units, similar to the one in Figure 3.10.

The screenshot shows the 'New Member' data entry form. It contains various input fields for member information: Image (with a 'Choose File' button), Member Id (with a required field note and examples), Member Type (a dropdown menu), Grade (a dropdown menu), First Name, Last Name, Address (with a required field note), Email (with a required field note and example), Telephone (with an example), Guardian Mobile (with an example), Gender (a dropdown menu), and Date of Birth (with a format note). At the bottom right, there are four buttons: Save, Edit, Clear, and View.

Figure 3.10: "New Member" data entry form

3.6.4 Data View Form

The system lets the user to view entered data for each module unit. Data view form provides facility to edit or delete selected data record, limit the number of display records, and go through the n set of records, export data and search specific data according to the users wish. Figure 3.11 shows data view form.

| Member Id | Grade | First Name | Address | Email | Telephone | Guardian Mobile | Picture | Action |
|-----------|-------|------------|--|---------------------------|------------|-----------------|---------|--------|
| ADM01 | Other | Ruvini | No. 343/1, Main Street, Ingiriya | ruvinidilrukshi@gmail.com | 0342268224 | 0714977086 | | |
| STU21106 | 10-A | Thanuga | No. 168, Pannila Kanda, Kananvila, Horana | thanugach@gmail.com | 0342261522 | 0715573711 | | |
| STU21990 | 10-A | Dineth | No. 37/01, Dewana Adiyara, Pinnakola Hena, Walpita, Horana | dinethsa@gmail.com | 0342261536 | 0775675041 | | |
| STU21996 | 10-A | Ishan | Minipura, Dumbara, Rathnapura | isdiluksha@yahoo.com | 0456352325 | 0714914523 | | |
| STU21999 | 10-A | Janith | No. 321/A, Minuwanwila Road, Kahatuduwa, Polgasowita | jaabishek@gmail.com | 0112365236 | 0779123016 | | |

Figure 3.11: “User Information” data view form

3.6.5 Member Search Form

Figure 3.12 shows “Member Search” form.

Search Member by "Membership Number"

Search Member by "Name"

Figure 3.12: “Member Search” form

By using “Search Member by ‘Membership Number’” panel, we can get user information with check-out and holdings, according to the entered membership number. Figure 3.13 shows member information with details form.

CHAPTEY 04 – IMPLEMENTATION

4.1 Introduction

The implementation phase of software development is the process of converting a system requirements specification into an executable software system. In this phase software is developed in order to satisfy the all of the requirements identified during the analysis phase.

When coding the system, using comments in appropriate places is very important. It enables the source code easier to understand and very useful to rework the code and it is very useful to another party who reviews the code. Validation is also an important thing to consider while we programming.

4.2 Implementation Environment

| Hardware Requirements | |
|---|--|
| Intel (R) core (TM) i5-5210U CPU @ 1.7 GHZ 2 GB Random Access Memory (Minimum) 60 GB or higher Hard Disk (Minimum) Printer Barcode Reader | |
| Software Requirements | |
| Documentation Microsoft Word 2010 Acrobat Reader 8.0 Argo UML v0.32.1 | Design Tools (Front end) Adobe Photoshop CS3 |
| Developing Tools XHTML CSS JS Pure PHP Net Beans IDE 8.0.2 My SQL Workbench 6.2 CE Bootstrap Framework | Operating System Windows XP or higher |

Table 4.1: Implementation details

4.3 System Developed Tools and Technologies

The following tools and technologies were used to development process activities of the system.

My SQL Workbench - Used to handle database and diagraming. MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. My SQL Workbench 6.2 CE is used to develop the System. [11]

Argo UML - Used for diagraming. ArgoUML is an UML diagramming application written in Java and released under the open-source Eclipse Public License. Argo UML v0.32.1 is used to develop the System. [12]

Net Beans IDE - NetBeans is a software development platform written in Java and runs on Microsoft Windows, macOS, Linux, Solaris and other platforms supporting a compatible JVM. It is primarily intended for development in Java, but also supports other languages, in particular PHP, C/C++ and HTML5. Net Beans IDE 8.0.2 is used to develop the System. [13]

XAMPP - XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, stands for Cross-Platform (X), Apache (A), MySQL database (M), PHP (P) and Perl (P). It is a simple, lightweight and that makes it extremely easy for developers to create a local web server for testing and deployment purposes. XAMPPv3.2.1 is used to develop the System. [14]

Bootstrap - Bootstrap is a free and open source front end web framework for designing websites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. [15]

Adobe Photoshop CS3 – Adobe Photoshop is a raster graphic editor which used for image retouching.

PHP - Server side scripting language used to develop the main system and its logics.

XHTML – Markup language which used for basic interfaces building of the system.

CSS - Used to make the plain XHTML interfaces more attractive and user friendly which also define styles for the system.

JavaScript - Used for building client side validations.

JQuery - Based on JavaScript and used to implement pre-coded modules.

MySQL - Open-source relational database management system (RDBMS) used to handle all the database related activities.

Ajax - Ajax is a JavaScript based technology and it supports updating the system components without refreshing the whole system page.

4.4 Code Features

Three-tier architecture was used for this system. Figure 4.1 shows combination of these three layers.

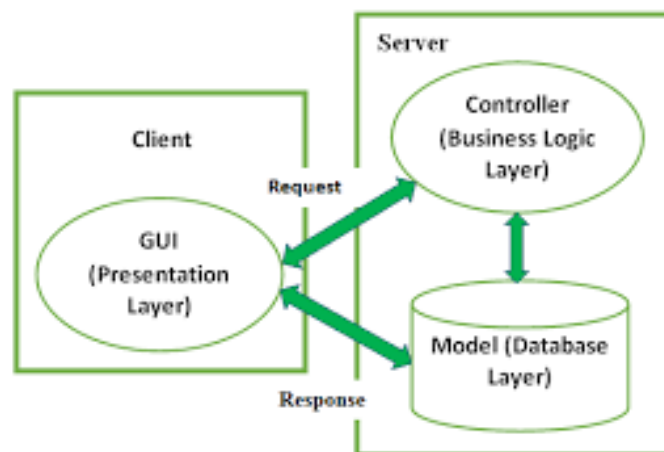


Figure 4.1: Three-tier architecture

MVC architecture was used for this system. MVC stand for the three layers named as Model, View and Control. All the connections of the database are stored in the model layer. All the views of the system which can be seen to the end user are stored in the view layer. All the programing scripts are stored in the control layer. It is worth to maintain the connectivity among these three layers to work the system effectively and efficiency. Following figure 4.2 shows the directory structure of the system.

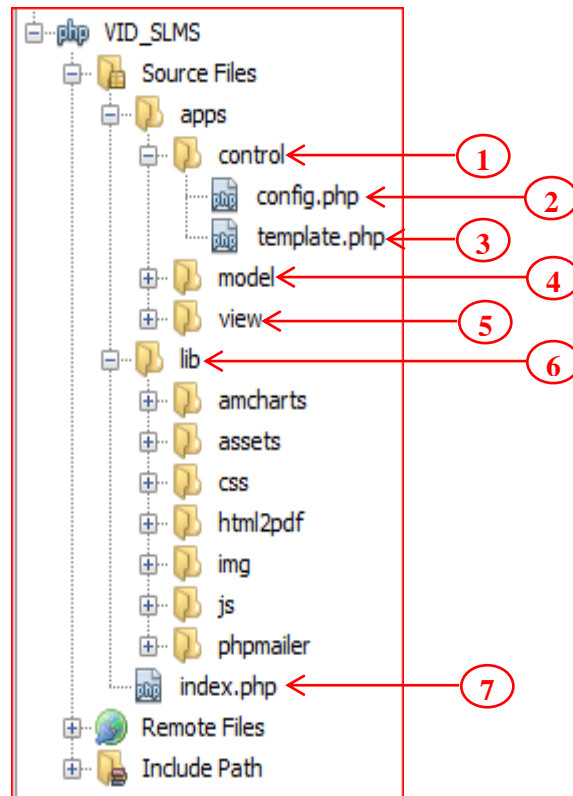


Figure 4.2: Directory Structure

1 → 'control' folder → contains config.php and template.php files.

2 → 'config.php' file → contains configuration level variables. (Database and site configuration)

3 → 'template.php' file → contains class functions.

4 → 'model' folder → stores basic entities (catalog, circulation, item management, user management, acquisition and report) used by the application, usually from a database and contain the business logic of the application.

5 → 'view' folder → contains all the user interfaces of the application.

6 → 'lib' folder → contains css, js, images and library files.

7 → 'index.php' file → home page of the application.

4.4.1 Database Connection Page

After the database connection established successfully, a database server and its client software can communicate with each other. The system always connects with the

particular database using database connection on add, edit or delete mode. MySQL function called “PDO” is used to manage the connection by passing three parameters namely server name, user name, and the password. The following code fragment shows the code for database connection.

The PHP Data Objects (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP. Each database driver that implements the PDO interface can expose database-specific features as regular extension functions. Note that you cannot perform any database functions using the PDO extension by itself; you must use a database-specific PDO driver to access a database server. [16]

```
<?php

// CREATE NEW SESSION
session_start();

// SET TIMEOUT
$inactive = 6000;
if (!isset($_SESSION['timeout']))
    $_SESSION['timeout'] = time() + $inactive;
$_SESSION['life'] = time() - $_SESSION['timeout'];

if ($_SESSION['life'] > $inactive) {
    session_destroy();
    header("Location:../../index.php");
}
$_SESSION['timeout'] = time();

// DATABASE CONFIGURATION
define('DB_HOST', 'localhost');
define('DB_PORT', '3306');
define('DB_NAME', 'db_slms');
define('DB_USER', 'root');
define('DB_PASS', '');

// DATABASE CONNECTION-STRING
$conn = new PDO('mysql:host=' . DB_HOST . ';port=' . DB_PORT . ';dbname=' . DB_NAME, DB_USER, DB_PASS);

// ERROR HANDLING
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// SITE CONFIGURATION
define('BASE_URL', 'http://localhost/VID_SLMS/');
define('LOCAL_PATH', 'C:/xampp/htdocs/VID_SLMS/');
define('SITE_TITLE', 'Vidyarathna University College | Library Automation System');

// SET TIMEZONE
date_default_timezone_set('Asia/Colombo');

require_once 'template.php';
?>
```

4.4.2 Login Handling Page

```
<?php
require_once '../control/config.php';

if (isset($_POST['LOGIN'])) {

    $login_userName = $_POST['login_userName']; // user name as library registration number
    $login_password = sha1($_POST['login_password']); // encrypted password

    try {
        // entered user name and password tally with the DB and check the user status & expDate
        $sql = "SELECT login_userName, login_password, login_type, user_pic, user_expDate FROM tbl_login, tbl_user
                WHERE login_userName=user_memberId and user_status='Active'
                and login_userName=:login_userName and user_expDate > CURDATE()";

        $stmt = $conn->prepare($sql);
        $stmt->execute(array(':login_userName' => $login_userName));
        $result = $stmt->fetchAll();

        // check user status and validity
        if (count($result)) {
            $row = $result[0];

            // assign DB password to variable
            $login_passwordDB = $row['login_password'];

            // check DB password with user entered password
            if ($login_passwordDB == $login_password) {

                // assign username, login type and user pic to session
                $_SESSION['username'] = $row['login_userName'];
                $_SESSION['login_type'] = $row['login_type'];
                $_SESSION['user_pic'] = $row['user_pic'];

                // success message & redirect to home page
                $_SESSION['SUCCESS'] = 'Welcome to Vidyarthna Library ';

                if ($_SESSION['login_type'] == 'Administrator') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeAdm.php');
                }
                if ($_SESSION['login_type'] == 'Librarian') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeLibrarian.php');
                }
                if ($_SESSION['login_type'] == 'Staff') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeStaff.php');
                }
                if ($_SESSION['login_type'] == 'Teacher') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeOther.php');
                }
                if ($_SESSION['login_type'] == 'Student') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeOther.php');
                }
            } else {

                // display if password is incorrect
                $_SESSION['ERROR'] = 'Please check your password!';
                header("Location: " . $_SERVER['HTTP_REFERER']);
            }
        } else {

            // display incorrect user or inactive user
            $_SESSION['ERROR'] = 'Please check your user name and password or invalid user!';
            header("Location: " . $_SERVER['HTTP_REFERER']);
        }
    } catch (Exception $ex) {
        $_SESSION['ERROR'] = $ex->getMessage();
        header("Location: " . $_SERVER['HTTP_REFERER']);
    }
}

?>
```

4.4.3 Preventing the login details

To secure the data of the login page SHA1 (Secure Hash Algorithm) is used to encrypt the password.

```
$login_userName = $_POST['login_userName']; // user name as library registration number
$login_password = sha1($_POST['login_password']); // encrypted password
```

4.4.4 Insert, Update and Delete Data

Following code shows how to handle insert, update and delete records in the system. Note that full codes are not included, because of the limited space.

Inserting a user record is done using following code fragment.

```
// if user press SAVE button execute below code and add new record to DB
if (isset($_POST['Save'])) {

    // upload user picture -----
    $allowedExts = array("gif", "jpeg", "jpg", "png");
    $temp = explode(".", $_FILES["user_pic"]["name"]);
    $extension = end($temp);

    if (true) {  
        $SESSION['ERROR'][] = "Invalid file";  
    }

    // upload user picture -----
    //===== Create User
    // insert SQL
    $sql = 'INSERT INTO tbl_user(user_memberId,user_grade,user_fname,user_lname,user_address,user_email,user_tel,'
        . 'user_guardianMob,user_gender,user_dob,user_pic,user_expDate,user_status) '
        . 'VALUES(:user_memberId,:user_grade,:user_fname,:user_lname,:user_address,:user_email,:user_tel,'
        . ':user_guardianMob,:user_gender,:user_dob,:user_pic,:user_expDate,:user_status)';

    // connecting to DB
    $stmt = $conn->prepare($sql);

    // pass values to DB
    $stmt->execute(array('user_memberId' => $user_memberId,
        'user_grade' => $user_grade,
        'user_fname' => $user_fname,
        'user_lname' => $user_lname,
        'user_address' => $user_address,
        'user_email' => $user_email,
        'user_tel' => $user_tel,
        'user_guardianMob' => $user_guardianMob,
        'user_gender' => $user_gender,
        'user_dob' => $user_dob,
        'user_pic' => $user_pic,
        'user_expDate' => $user_expDate,
        'user_status' => $user_status));

    //===== Create Login
    // insert SQL
    $sqlL = 'INSERT INTO tbl_login(login_userName,login_password,login_type) '
        . 'VALUES(:login_userName,:login_password,:login_type)';

    // connecting to DB
    $stmtL = $conn->prepare($sqlL);

    // pass values to DB
    $stmtL->execute(array('login_userName' => $login_userName,
        'login_password' => $login_password,
        'login_type' => $login_type));

    // after success disconnect from DB
    $conn = null;

    // print success msg
    $SESSION['SUCCESS'] = 'Created user ' . ' ' . $user_memberId . ' ' . ' successfully';
}
```

Updating a user record is done using following code fragment.

```
// if user press EDIT button (After SAVE the record), execute below code and edit the current record on the DB
if (isset($_POST['Edit'])) {

    //===== Edit User
    // update SQL
    $sql = 'UPDATE tbl_user '
        . 'SET user_grade=:user_grade, user_fname=:user_fname, user_lname=:user_lname, user_address=:user_address, '
        . 'user_email=:user_email, user_tel=:user_tel, user_guardianMob=:user_guardianMob, user_gender=:user_gender,'
        . 'user_dob=:user_dob WHERE user_memberId=:user_memberId';

    // connecting to DB
    $stmt = $conn->prepare($sql);

    // pass values to DB
    $stmt->execute(array('user_memberId' => $user_memberId,
        'user_grade' => $user_grade,
        'user_fname' => $user_fname,
        'user_lname' => $user_lname,
        'user_address' => $user_address,
        'user_email' => $user_email,
        'user_tel' => $user_tel,
        'user_guardianMob' => $user_guardianMob,
        'user_gender' => $user_gender,
        'user_dob' => $user_dob));

    //===== Edit Login
    // update SQL
    $sqlL = 'UPDATE tbl_login '
        . 'SET login_userName=:login_userName, login_password=:login_password, login_type=:login_type'
        . 'WHERE login_userName=:login_userName';

    // connecting to DB
    $stmtL = $conn->prepare($sqlL);

    // pass values to DB
    $stmtL->execute(array('login_userName' => $login_userName,
        'login_password' => $login_password,
        'login_type' => $login_type));

    // print success msg
    $_SESSION['SUCESSS'] = 'Updated user ' . '' . $user_memberId . '' . ' successfully';
}
```

Deleting a user record is done using following code fragment.

```
//===== Delete User
// delete SQL
$sql = 'DELETE FROM tbl_user '
    . 'WHERE user_memberId=:user_memberId';

// connecting to DB
$stmt = $conn->prepare($sql);

// execute query in DB
$stmt->execute(array('user_memberId' => $user_memberId));

//===== Delete Login
// delete SQL
$sqlL = 'DELETE FROM tbl_login WHERE login_userName=:login_userName';

// connecting to DB
$stmtL = $conn->prepare($sqlL);

// execute query in DB
$stmtL->execute(array('login_userName' => $login_userName));

// print success msg
$_SESSION['SUCESSS'] = 'Removed user ' . '' . $user_memberId . '' . ' successfully';
```

4.4.5 Data Validation

Data validation is the process of ensuring that a program operates on clean, correct and useful data. It uses routines, often called "validation rules" "validation constraints" or "check routines", that check for correctness, meaningfulness, and security of data that are input to the system.

Data validation can be done as client side validation, server side validation, session validation and database validation.

Client side validation

Client side validation is performed on the client machine's web browser. In this system, client side validation is carried out by HTML5 and JQuery coding.

Following coding shows how the validating of the Email is carried out in the user registration form. Inside the 'input' element, 'type' attribute denotes that the type is 'email' and 'required' attribute denotes that the email is required. Using the 'span' tag, it is informed that the email is required and the example of the email also given.

```
<div class="form-group">
  <label class="col-md-3 control-label">Email:</label>
  <div class="col-md-9">
    <input type="email" class="form-control" name="user_email" required="" value="php
      if (isset($_GET['user_memberId'])) {
        echo $user_email;
      }
    ?" />
    <span class="help-block">required email</span>
    <span class="help-block">Example: example@gmail.com</span>
  </div>
</div>
```

Following coding shows that the telephone number is validated in the user registration form. Inside the 'input' element, 'type' attribute denotes that the type is 'text' and 'pattern' attribute denotes that the telephone number starts with zero, next with one to nine and remainder fills with numbers which zero to nine and telephone number limited to ten digits. 'Title' attribute denotes the helping message. Using the 'span' tag, it is given that the example of the telephone number.

```

<div class="form-group">
  <label class="col-md-3 control-label">Telephone:</label>
  <div class="col-md-9">
    <input type="text" class="form-control" name="user_tel" pattern="0[1-9]{1}[0-9]{8}"
      title="Ten digit number start with zero" value=<?php
        if (isset($_GET['user_memberId'])) {
          echo $user_tel;
        }
      ?> >
    <span class="help-block">Example: 0814563217</span>
  </div>
</div>

```

PHP validation

PHP validation is a can be categorized as server side validation. If session success, it displays the message “Welcome to Vidyarathna Library”.

```

// check DB password with user entered password
if ($login_passwordDB == $login_password) {

    // assign username, login type and user pic to session
    $_SESSION['username'] = $row['login_userName'];
    $_SESSION['login_type'] = $row['login_type'];
    $_SESSION['user_pic'] = $row['user_pic'];

    // success message & redirect to home page
    $_SESSION['SUCESSS'] = 'Welcome to Vidyarathna Library ';
}

```

Errors are identified using “try” and “catch”.

```

// To handle errors
try {

    // if user press SAVE button execute below code and add new record to DB
    if (isset($_POST['Save'])) {

        //===== Create User
        // insert SQL
        $sql = 'INSERT INTO tbl_user(user_memberId,user_grade,user_fname,user_lname,user_address,user_email,user_tel,'
            . 'user_guardianMob,user_gender,user_dob,user_pic,user_expDate,user_status) '
            . 'VALUES(:user_memberId,:user_grade,:user_fname,:user_lname,:user_address,:user_email,:user_tel,'
            . ':user_guardianMob,:user_gender,:user_dob,:user_pic,:user_expDate,:user_status)';
    }
}

```

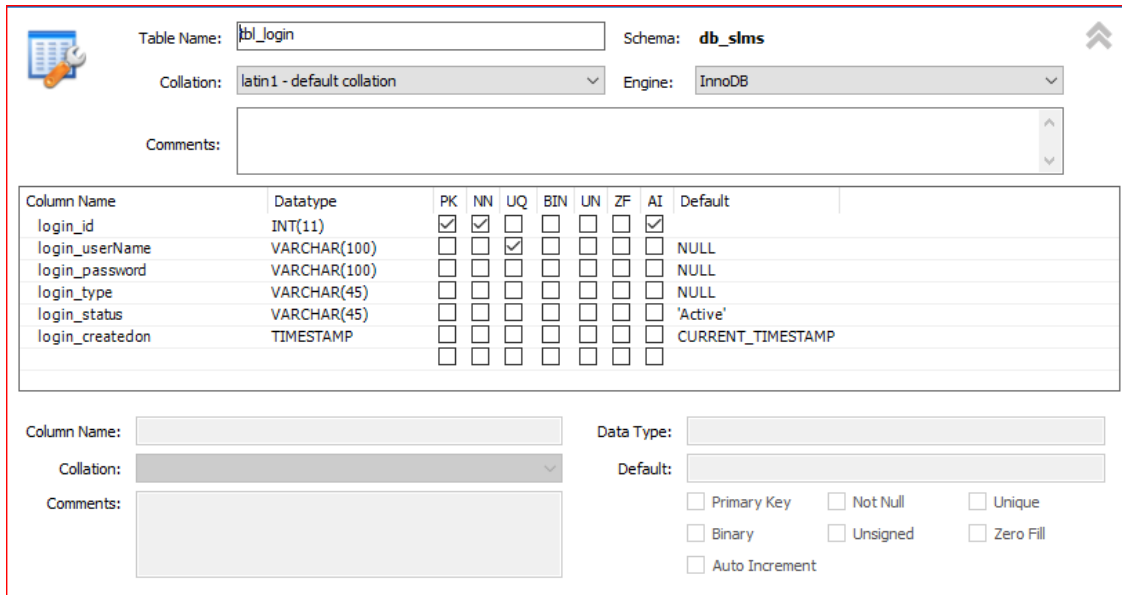
```

// Print error msg while ...
catch (Exception $ex) {
    if ($ex->getCode() == '23000') {
        $_SESSION['ERROR'] = ' User Member Id ' . '' . $user_memberId . '' . ' already exists ... try another !!!';
    } else {
        $_SESSION['ERROR'] = $ex->getMessage();
    }
}

```

Database Validation

In each and every table, important fields are validated. Tbl_login validated using primary key, not null, auto increment, unique key, assign default login status and assign system date to default 'login_createdon' date.



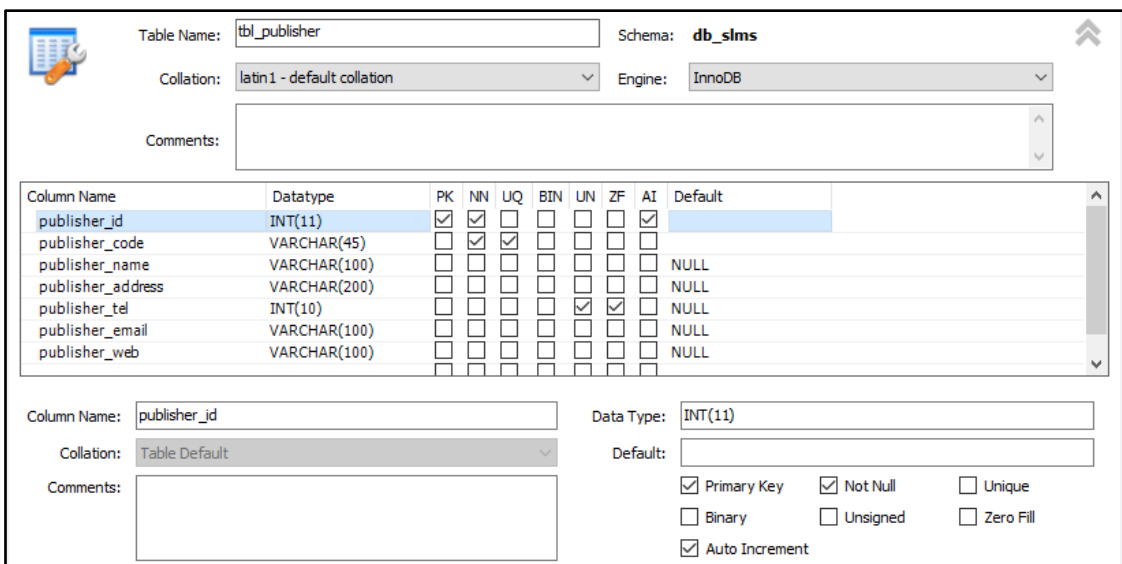
The screenshot shows the MySQL Workbench Table Designer for the table `tbl_login` in the `db_slms` schema. The table is using the `latin1 - default collation` and the `InnoDB` engine. The columns are defined as follows:

| Column Name | Datatype | PK | NN | UQ | BIN | UN | ZF | AI | Default |
|-----------------|--------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|-------------------|
| login_id | INT(11) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | |
| login_userName | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| login_password | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| login_type | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| login_status | VARCHAR(45) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | 'Active' |
| login_createdon | TIMESTAMP | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | CURRENT_TIMESTAMP |

The bottom section shows the configuration for a new column:

Column Name:
Collation:
Comments:
Data Type:
Default:
☐ Primary Key ☐ Not Null ☐ Unique
☐ Binary ☐ Unsigned ☐ Zero Fill
☐ Auto Increment

Tbl_publisher validated using primary key, not null, auto increment, unique key, unsigned and zero fill.



The screenshot shows the MySQL Workbench Table Designer for the table `tbl_publisher` in the `db_slms` schema. The table is using the `latin1 - default collation` and the `InnoDB` engine. The columns are defined as follows:

| Column Name | Datatype | PK | NN | UQ | BIN | UN | ZF | AI | Default |
|-------------------|--------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---------|
| publisher_id | INT(11) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | |
| publisher_code | VARCHAR(45) | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | |
| publisher_name | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| publisher_address | VARCHAR(200) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| publisher_tel | INT(10) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | NULL |
| publisher_email | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |
| publisher_web | VARCHAR(100) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | NULL |

The bottom section shows the configuration for a new column:

Column Name:
Collation:
Comments:
Data Type:
Default:
☒ Primary Key ☒ Not Null ☐ Unique
☐ Binary ☐ Unsigned ☐ Zero Fill
☒ Auto Increment

Session Validation

The following code shows check login type using session.

```
// check DB password with user entered password
if ($login_passwordDB == $login_password) {

    // assign username, login type and user pic to session
    $_SESSION['username'] = $row['login_userName'];
    $_SESSION['login_type'] = $row['login_type'];
    $_SESSION['user_pic'] = $row['user_pic'];

    // success message & redirect to home page
    $_SESSION['SUCESSS'] = 'Welcome to Vidyarathna Library ';

    if ($_SESSION['login_type'] == 'Administrator') {
        header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeAdm.php');
    }
    if ($_SESSION['login_type'] == 'Librarian') {
        header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeLibrarian.php');
    }
    if ($_SESSION['login_type'] == 'Staff') {
        header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeStaff.php');
    }
    if ($_SESSION['login_type'] == 'Teacher') {
        header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeOther.php');
    }
    if ($_SESSION['login_type'] == 'Student') {
        header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeOther.php');
    }
} else {

    // display if password is incorrect
    $_SESSION['ERROR'] = 'Please check your password!';
    header("Location: " . $_SERVER['HTTP_REFERER']);
}
```

4.5 Re-Usable Components

The following re-usable components have been used when implementing the system to reduce development time, quality improvement, and easy maintenance and to maximize the efficiency of the system.

- Boostarp Atalnt admin template.
- Used CSS Re-usable Components.
- Dropdown menu code has been used when styling dropdowns.
- Form controllers code has been used when styling forms.
- Form field styling code has been used when styling forms.
- Date Picker has been used when designing date selection controllers.
- Slide Toggle code has been used when applying styling to interfaces.
- Data table code has been used when adding data tables.

- Pagination plugin has been used to manage data contents inside the datatable.
- Data Export code has been used when adding export button.
- Clock and date plugin.
- HTML2PDF used to implement reports.
- Amcharts used to implement charts.

CHAPTER 5 - EVALUATION

5.1 Introduction

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. [17]

Software testing is a verification and validation process which should be done during the development process. Verification is process which confirming that the system meets all the functionality. Validation is the process which assures that the product satisfies the specified user requirements at the end of the software development phase.

5.2 Software Testing Methods

There are two major techniques of software testing.

Black-Box Testing

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. Black-box testing is also called functional testing or behavioral testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. [17]

White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called glass testing or open-box testing. In order to perform white-box testing on an application, a tester needs to know the internal workings of the code. [17]

5.3 Types of Testing

5.3.1 Unit Testing

While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free. [17]

5.3.2 Integration Testing

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. (Eg: argument passing, data updating) [17]

5.3.3 System Testing

The software is compiled as product and then it is tested as a whole. This can be accomplished using functionality testing or performance testing or security & portability testing. [17]

5.3.4 Acceptance Testing

When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.

5.3.5 Regression Testing

Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code.

5.4 Test Plan and Test Cases

Test plan is concerned with scheduling and resourcing all of the activities in the testing process. It involves defining the testing process, taking into account the people and the time available. Usually, a test plan will be created, which defines what is to be tested, the predicted testing schedule, and how tests will be recorded. [7] Table 5.1 shows high level test plan of the library automation system.

| Module Name | Function Name | Test Priority |
|----------------------|-------------------------------|---------------|
| Member Management | Add member | High |
| | View member | High |
| | Search member | Medium |
| | Edit member | High |
| | Delete member | High |
| | Confirm deletion of member | Low |
| | Member login | High |
| Item Management | Add item | High |
| | View item | High |
| | Search item | Medium |
| | Edit item | High |
| | Delete item | High |
| | Confirm deletion of item | Low |
| Item copy management | Search item | High |
| | View item information | High |
| | View item copy information | High |
| | Create item copy | High |
| | Save item copy | High |
| | Edit item copy | High |
| Check-out item | Search item for check-out | High |
| | Check member validity | High |
| | Check check-out history | High |
| | Check overdue and fine | Medium |
| | Check-out item | High |
| | Delete check-out | High |
| | Confirm deletion of check-out | High |

| | | |
|-----------------|---|--------|
| Hold item | Search item for hold | High |
| | Check member validity | High |
| | Check holding history | High |
| | Hold item | High |
| | Delete holdings | High |
| | Confirm deletion of holdings | High |
| Check-in item | View item status | High |
| | Edit item status | High |
| | View edited item status | Medium |
| | Delete the check-out record from check-out list of particular member. | Medium |
| | Confirm deletion of check-out record | Medium |
| Reports | Generate daily/ weekly/ monthly check-out list | High |
| | Generate daily/ weekly/ monthly holding list | High |
| | Generate daily/ weekly/ monthly check-in list | High |
| | Generate daily/ weekly/ monthly/ annually member registration list | High |
| | Generate member history reports | High |
| | Generate famous check-out list by title/ author | High |
| | Generate statistic report | High |
| Messages | View email messages | High |
| | View recommended request | High |
| Supporting data | Add data | High |
| | View data | High |
| | Search data | High |
| | Edit data | High |
| | Delete data | High |
| | Confirm deletion of data | Low |

| | | |
|---------|---|------|
| Finance | Calculate fine | High |
| | Calculate registration fee | High |
| | Calculate renewal fee | High |
| Catalog | Select option for search | High |
| | Type text according to the selected option and search | High |
| | View search results | High |

Table 5.1: High level test plan

In the testing process, creating test cases is the most important. It consist description of test case, expected result and steps. The test cases were written for each module separately. Test cases are described at the following section. Please refer Appendix E for other test cases.

User Authentication

| No | Test Description | Steps to Test | Expected Result |
|----|-----------------------------|--|--|
| 1. | Validate user Input details | Enter correct user name and password | Dislplays “Welcome to Vidyarathna Library” and Direct to home page |
| 2. | Validate user Input details | Enter correct user name and incorrect password | Displays “Please check your password!” and load login page again. |
| 3. | Validate user Input details | Enter incorrect user name and correct password | Displays “Please check your user name & password!” |
| 4. | Validate user Input details | Enter both user name and pass word incorrectly | Displays “Please check your user name & password!” |

Table 5.2: User authentication

Member Registration Module

| No | Test Description | Steps to Test | Expected Result |
|----|-------------------------------|---|---|
| 1. | Add member details | Enter details correctly | Displays “Created member <member id> successfully” |
| 2. | Add member details | Save without entering data | Displays “Please fill out this Field” |
| 3. | Add member details | Enter invalid member id | Display “Please match the requested format” |
| 4. | Add member details | Enter incorrect email address | Displays “Please include an ‘@’ in the email address” |
| 5. | Add member details | Enter incomplete email address with ‘@’ sign | Displays “Please enter a part following ‘@’” |
| 6 | Add member details | Enter incomplete email address with ‘@’ and ‘.’ | Display “.’ Is used at a wrong position” |
| 7 | Validate member input details | Viewing the entered data of the members | Redirect to the “Member Information” page |
| 8 | Delete member record | Delete selected record | Display “Removed member <member id> successfully” and redirect to the “New Member” form. |
| 9 | Edit member record | Viewing the selected member details | Redirect to the “New Member” page on edit mode. After successfully edit the record, display “Updated member <member id> successfully” message |

Table 5.3: Member registration module

Check-out Module

| No | Test Description | Steps to Test | Expected Result |
|----|----------------------------|-----------------------------|-------------------------------------|
| 1 | Search member for checkout | Enter correct card number | Display member information |
| 2 | Search member for checkout | Enter incorrect card number | Display “Enter correct card number” |

| | | | |
|----|--------------------------|-------------------------|--|
| 3 | Validate entered barcode | Enter correct barcode | Check availability of entered barcode. If it is available, check member status. |
| 4 | Validate member | Enter correct barcode | If the member is active, check fine status. |
| 5 | Validate member | Enter correct barcode | If the member without fine, check current check-outs. |
| 6 | Validate member | Enter correct barcode | If the member is not exceeding the check-out limit, checked-out the selected item. Display "Check-out item successfully" |
| 7 | Validate member | Enter correct barcode | If the member is exceeding check-out limit, display "Exceed your check-out limit" |
| 8 | Validate member | Enter correct barcode | If the member with fine, display "Pay fine" |
| 9 | Validate member | Enter correct barcode | If the member is in-active, display "User expired. can't check-out" |
| 10 | Validate entered barcode | Enter correct barcode | Check availability of entered barcode. If it is not available, display "Item not available" |
| 11 | Validate entered barcode | Enter incorrect barcode | Display "Enter correct barcode" |

Table 5.4: Check-out module

Check-in Module

| No | Test Description | Steps to Test | Expected Result |
|----|-------------------------|----------------------------|---|
| 1 | Search check-out record | Enter item barcode | Display check-out details, member information and fine if any |
| 2 | Search check-out record | Enter invalid item barcode | Display "Not a valid barcode" |

| | | | |
|---|---|---|--------------------------------------|
| 3 | Verify member details, and pay fine, if there is. | Verify member details and click “Pay Fine” button | Display “Paid” |
| 4 | Verify member details, release check-out record and change the status of the item | Verify member details and click “Check In” button | Display “Item successfully check-in” |

Table 5.5: Check-in module

5.5 User Evaluation

User evaluation is done by selecting clients of the system. In this Library Automation System, the librarian has been taken as the highest privilege user, library staff has been taken as middle privilege users and other users (teachers and students) have been taken as normal privilege users of the system for testing. User evaluation questionnaire was given to target user group and results has been summarized. Figure 5.1 shows User Evaluation Questionnaire.

Library Automation System for Vidyarthna University College, Horana

User Evaluation Questionnaire

| Name of User: <i>Lakshman Kumar Aluthge</i> | | Role of User: <i>Librarian</i> | | | |
|---|-----------|--------------------------------|---------|------|-----------|
| Evaluating Item | Very Good | Good | Average | Poor | Very Poor |
| Overall reaction | | ✓ | | | |
| Character readability | ✓ | | | | |
| Color scheme | | | ✓ | | |
| System navigation | | ✓ | | | |
| Ease of usage | ✓ | | | | |
| Functionalities | ✓ | | | | |
| Interfaces | ✓ | | | | |
| Ease of learning | | ✓ | | | |
| Response time | | ✓ | | | |
| Comments | — | | | | |

Figure 5.1: User evaluation questionnaire

Summarized result of user evaluation of the system shows on figure 5.2

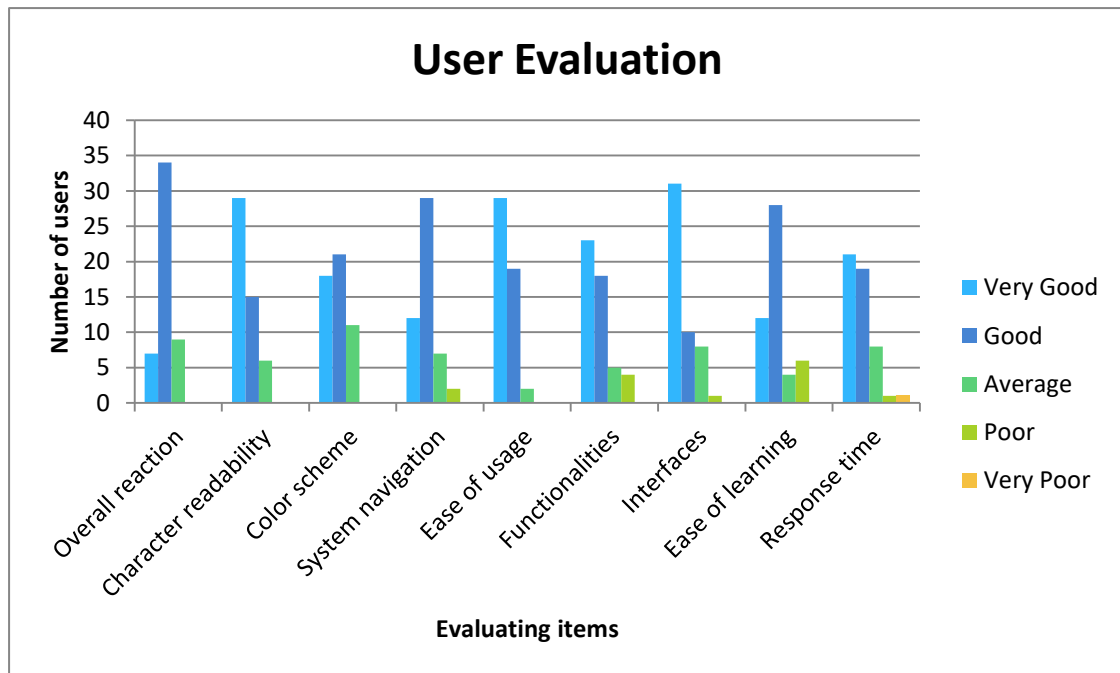


Figure 5.2: Summarized result of user evaluation of the system

The client certification has been attached to the Appendix.

CHAPTER 06 – CONCLUSION

6.1 Introduction

Library of the Vidyarthna University College, Horana has been selected to complete the third year software development project in the BIT degree course. Library Automation System is the theme of the project. It is a system which helps to do methodical work load of the school library. It designs to automate the librarian's activities. In that system, librarian can act as an administrator or librarian or both roles. Administrator, librarian and staff can handle item details and member related details. All of the actors of the system (administrator, librarian, staff, teacher, and student) can login to the system using their user name and password. Teacher and student can search items for check-outs, check their check-out history and request for new items. Librarian and staff can evaluate the reports.

Renew notifications, fine notifications, over dues and hold notifications are informed to the users via email. All the users of the system can update their profiles. So that most current information is available in the users table.

When developing the system, it was a great opportunity for me to gain very good knowledge. Lots of conflicts were occurred during development of the system. By solving one by one I was able to develop the system which can be satisfied by the client. This system will help the school to do the library activities effectively by saving time and reducing lot of paper work.

6.2 Future Improvements

The all of the requirements that are requested by the client is successfully implemented on the project. But the library automation system can be improved and upgraded into much more stable state by adding following steps.

Improve the message system

Set up alert to guardians to inform about member renewal, fine calculation and overdue items.

Add more reports

According to the needs of the library, generating more reports is essential to get to know about the performance of the library users.

Adding content pages of the item

Provide online login facility

Provide access to parents

6.3 Lessons Learnt

As the third year project, developing a system is somewhat hard, because it was required to apply all the theories which learnt in the previous years, practically. Library automation system had to be set up the library and had to understand them to convert in to an automated library system.

The system was developed using the server side scripting language, PHP, and there are lot of things to learn and identified lot of function. Developing a system, step by step while writing notes, was a good experience as a new developer to make aware of the system. And always reviewing new details and studying about other systems were helped to gain knowledge. Working with a real world client, develops the confidence to make the system effectively. Therefore this project was improved my technical, communicational and other soft skills which are mainly helping for the future carrier path in the field of information technology.

REFERENCES

- [1] School library [Online]. Available: http://www.abc-clio.com/ODLIS/odlis_s.aspx [Accessed: 31 - March - 2017].
- [2] Library automation [Online]. Available: http://www.abc-clio.com/ODLIS/odlis_1.aspx [Accessed: 31 - March - 2017].
- [3] Jeffrey L Whitten & Lonnie D Bentley, *Systems Analysis and Design methods*, 7th edition, 2007
- [4] *Koha* library automation system [Online]. Available: <https://koha-community.org/about/> [Accessed: 19 - April - 2017]
- [5] *Alice* for windows library automation system [Online]. Available: <https://www.softlinkint.com/> [Accessed: 19 – April - 2017]
- [6] CDS/ISIS for windows [Online]. Available: http://www.unesco.org/new/en/kathmandu/about-this-office/single-view/news/newly_opened_transitional_justice_resource_centre_uses_unesc/ [Accessed: 19 – April - 2016]
- [7] Ian Sommerville, *Software Engineering*, Addison-Wesley, 9th edition, 2011
- [8] System design [Online]. Available: https://en.wikipedia.org/wiki/Systems_design [Accessed: 20 – May - 2017]
- [9] Use case diagram [Online]. Available: <https://www.smartdraw.com/use-case-diagram/> [Accessed: 20 – May - 2017]
- [10] Class diagram [Online]. Available: https://en.wikipedia.org/wiki/Class_diagram [Accessed: 20 – May – 2017]
- [11] My SQL work bench [Online]. Available: https://en.wikipedia.org/wiki/MySQL_Workbench [Accessed: 25 – May – 2017]
- [12] Argo UML [Online]. Available: <https://en.wikipedia.org/wiki/ArgoUML> [Accessed: 25 – May – 2017]

- [13] Net Beans IDE [Online]. Available: <https://en.wikipedia.org/wiki/NetBeans> [Accessed: 25 – May – 2017]
- [14] XAMPP [Online]. Available: <https://en.wikipedia.org/wiki/XAMPP> [Accessed: 25 – May – 2017]
- [15] Bootstrap [Online]. Available: [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)) [Accessed: 25 – May – 2017]
- [16] PHP data object [Online]. Available: <http://php.net/manual/en/intro.pdo.php> [Accessed: 25 – June – 2017]
- [17] Software testing process [Online]. Available: https://www.tutorialspoint.com/software_testing/software_testing_quick_guide.htm [Accessed: 25 – June – 2017]

Appendix A – System Documentation

This documentation provides the Software and Hardware verifications, installation and information that are required to implement the system on the client site.

Hardware requirements and software requirements are shown in the Table A1 and Table A2

Hardware Requirements

| | |
|-----------------|----------------------------|
| Processor | 1.7 GHz Internal Processor |
| Memory | 2 GB |
| Hard Disk Drive | 60 GB |
| Display | 1366 x 768 resolution |
| Printer | Laser printer |
| Barcode reader | |
| Internet | ADSL Connection |

Table A1: Hardware requirements

Software Requirements

| | |
|------------------|--|
| Operating System | Windows 10 Pro 64-bit |
| XAMPP | Control Panel v3.2.1 XAMPP Version: 1.8.3 Apache 2.4 MySQL 3306 |
| Web Browser | Google chrome (Version 53.0.2785.116) |
| Image Editor | Adobe Photoshop CS3 |
| Code Editor | Net Beans IDE 8.0.2 |
| PDF Converter | Adobe Reader PDF |

Table A2: Software requirements

Installing XAMPP

Download XAMPP for Windows 64 bit operating system (refer Table A2 for the Minimum Version) from <http://www.apachefriends.org>. Install and install to the C:\ drive of the computer.

Installing web browser

Install browsers (refer Table A2 for the version and recommended browsers).

Files extraction

Open the CD and copy the VID_SLMS folder and paste it to the directory path "C:\xampp\htdocs".

Database installation

Open the web browser and type the URL <http://localhost/phpmyadmin> and enter Username and Password. Create empty database by providing name as "db_slms" and navigate to the "Import" tab and click "choose file" button. Then browse the CD and select the "VID_SLMS.sql" file by opening database folder.

Then Press "GO" button which located in the bottom of the page.

Network Access

If you wish to access this system via network, Change the "config.php" file as follows.

`define('DB_HOST', 'localhost');` ➔ `define('DB_HOST', <IP address>);`

Launching System

Verify the XAMPP is running, go to the "C:\xampp\" and open the "xampp-control.exe" file and verify whether Apache and MySQL are running. Open the installed web browser and type the URL "http://localhost/VID_SLMS" and press "Enter" button to access the system by providing correct username and password.

If you wish to access this system via network, open the installed web browser and type the URL "http://IP address/VID_SLMS" and press "Enter" button to access the system by providing correct username and password.

In the Appendix-C User Documentation is described to get the idea about how to operate the system.

Appendix B – Design Documentation

Use Case Diagrams and Descriptions

The following sub use case diagrams with descriptions might give a clear picture and understanding about the design of the system. To understand the main use case diagram of the system, easily and clearly it was categorized into seven sub modules.

Use Case Diagram for Member Registration

User makes request to registration. Administrator checks member availability. User uses their school registration number as the library registration number with three letter prefix. If the member is not available, details of the new member will be added to the database as a new member under the category of user type.

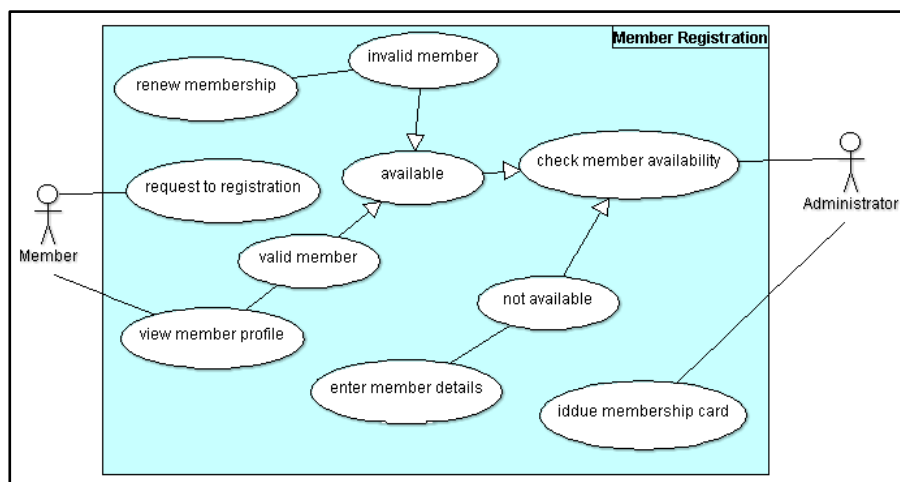


Figure B1: Member registration module

| | |
|----------------|---|
| Use case name | Member registration |
| Actors | Member, Administrator |
| Overview | Create and manage the new user Renew the invalid member View member details |
| Pre-condition | Create and manage new user or renew the invalid member |
| Description | Member should hand over the duly filled form with request to register the library. Administrator checks the availability of the member. If it is not available, add a new user. If it is available and valid member, member can view member profile. If it is available and invalid member, administrator can renew the membership. |
| Post condition | If success, add entry to the registration form or renew the member. If pop up an error message administrator has to fill the required fields. |

Table B1: Use case description for member registration module

Use Case Diagram for Item Management

First check availability of the new item by librarian or staff. If it is available, create a copy, otherwise add new item details to the item table and create a copy of the new item.

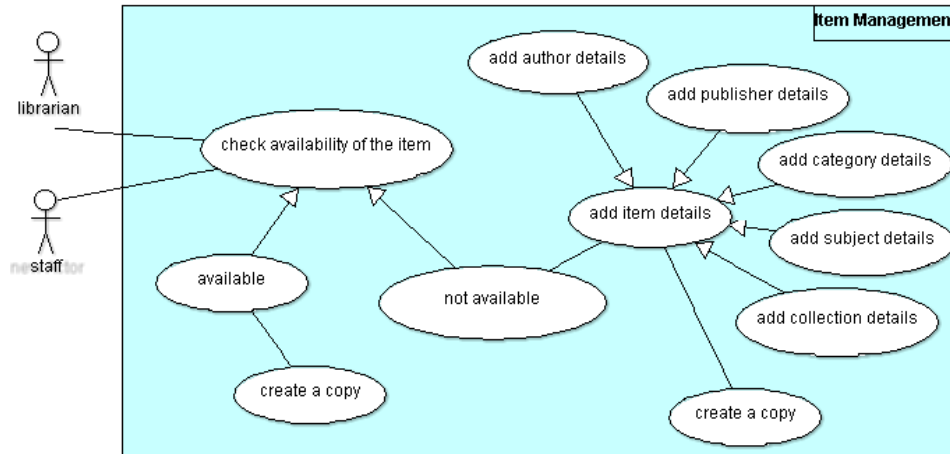


Figure B2: Item management module

| | |
|----------------|--|
| Use case name | Item management |
| Actors | Librarian, Staff |
| Overview | Check availability of the new item Create a copy of the item |
| Pre-condition | Check availability of an item and create a copy |
| Description | <p>Librarian or staff checks the availability of the new item.</p> <p>If the item is available, create a copy of the new item.</p> <p>If the item is not available, add item details.</p> <p>Add author details, if author of the new item is not available in the author table.</p> <p>Add publisher details, if publisher of the new item is not available in the publisher table.</p> <p>Add category details, if category of the new item is not available in the category table.</p> <p>Add subject details, if subject of the new item is not available in the subject table.</p> <p>Add collection details if collection of the new item is not available in the collection table.</p> <p>Create a copy of the item.</p> <p>Add location details, if location of the new item is not available in the location table.</p> |
| Post condition | <p>If success, create a copy of the new item and add entry to the item table and item copy table.</p> <p>If pop up an error message librarian or staff should get required action.</p> |

Table B2: Use case description for item management module

Use Case Diagram for Search Items for Check-out

Student or teacher or staff or librarian enters parameters for search items. If it is found the item librarian or staff login to the system and make check-out.

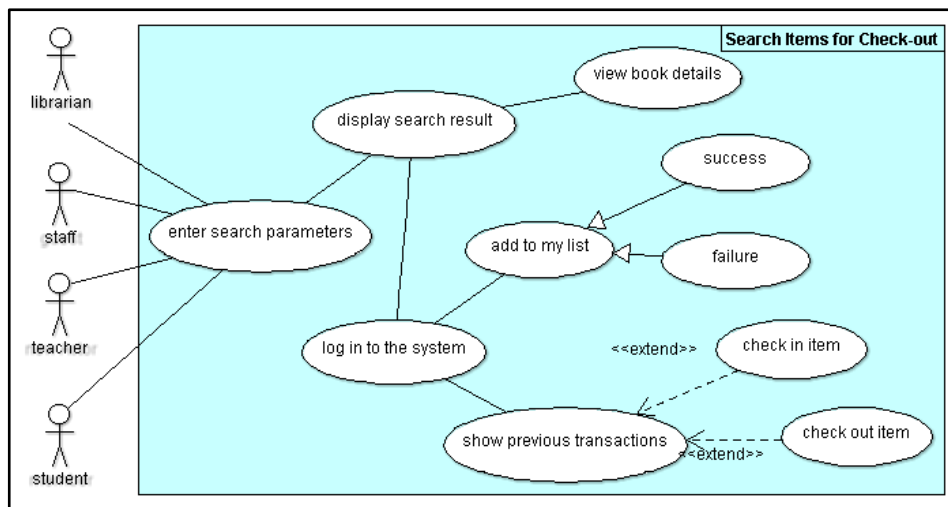


Figure B3: Search items for check-out module

| | |
|----------------|---|
| Use case name | Search items for check-out |
| Actors | Student, teacher, staff, librarian |
| Overview | Enter parameters for search items. |
| Pre-condition | Everyone can search items. But only members can check-out items. |
| Description | Enter parameters of search items. Display search results. Can view item details by clicking each search result. Member can select the item for check-out. Request to librarian or staff to make check-outs. |
| Post condition | If success, update check-out book's status and add entry to the check-out table. If pop up an error message librarian or staff should get required action. |

Table B3: Use case description for search items for check-out

Use Case Diagram for Check-out Items

Member makes request to check-out books. Librarian or staff logs in to the system and check member availability. If the member is available, check the member validity. If the member is valid, he can check-out books.

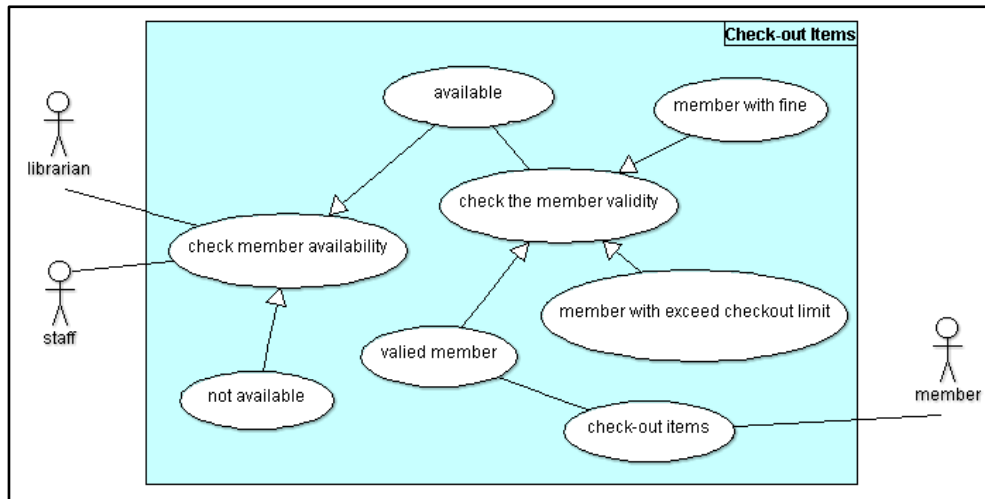


Figure B4: Check-out items module

| | |
|----------------|--|
| Use case name | Check-out items |
| Actors | Librarian, Staff, Member |
| Overview | Check member validity Check-out items |
| Pre-condition | Check the member validity and check-out items |
| Description | Member should request to check-out items. Librarian or staff checks the availability and validity of the member. If the member is available and valid he/ she can check-out items. If the member is available with fine he/ she can pays fine and check-out items. If the member is available with exceed checkout limit, he/ she can't check-out items. |
| Post condition | If success, update check-out item's status and add entry to the check-out table. If pop up an error message librarian or staff should get required action. |

Table B4: Use case description for check-out items

Use Case Diagram for Reserve an Item

Member searches item for check-out. If the item is available in the collection and someone checked-out, the requested member can reserve that item. If the item will available the requested member will inform by email.

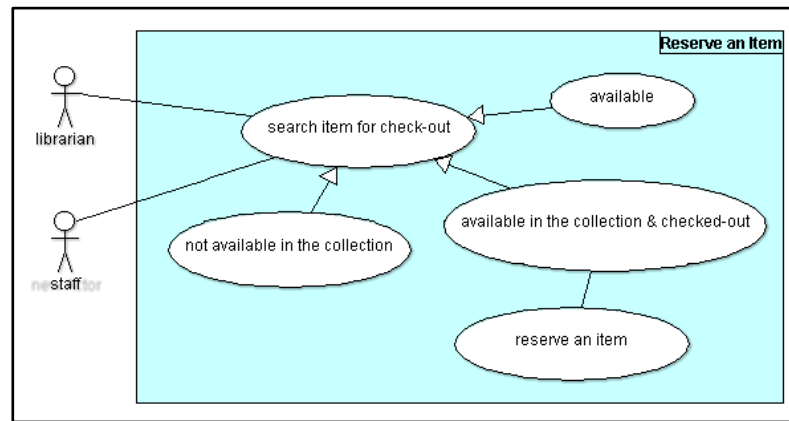


Figure B5: Reserve an item module

| | |
|----------------|--|
| Use case name | Reserve an item |
| Actors | Librarian, Staff |
| Overview | Search item for check-out. Reserve requests are valid for checked-out items. |
| Pre-condition | Check the item status and reserve it. |
| Description | Member search item for check-out. Select item for check-out. If the selected item was checked-out, Librarian or staff checks the availability and validity of the member. If the member is available and valid he/ she can reserve items. If the member is available with fine he/ she can pays fine and can reserve items. If the member is available with exceed check-out limit, he/ she can reserve items. If the member is available with exceed reserve limit, he/ she can't reserve items. |
| Post condition | If success, update checked-out item's status and add entry to the hold item table. If pop up an error message librarian or staff should get required action. |

Table B5: Use case description for reserve an item

Use Case Diagram for Check-in Item

Member makes request to check-in item. Librarian or staff enters item barcode on check-in window. If the member had to pays fine, pays fine and check-in item.

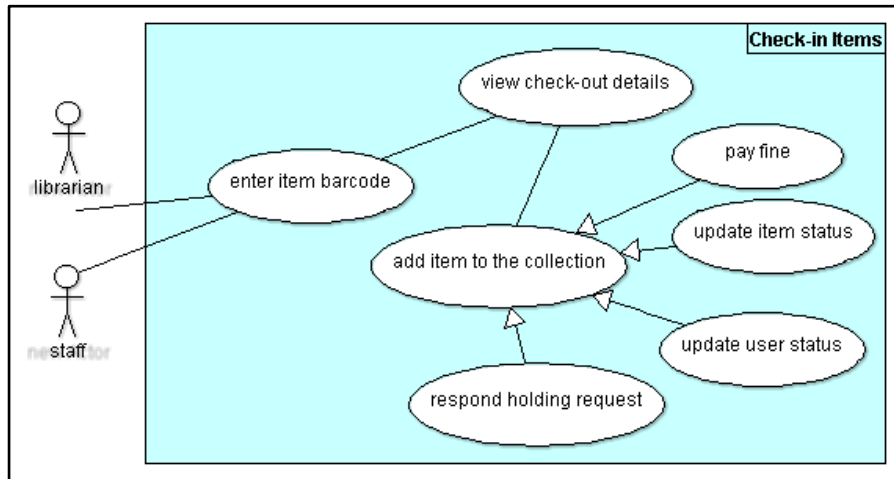


Figure B6: Check-in item module

| | |
|----------------|--|
| Use case name | Check-in item |
| Actors | Librarian, Staff, Member |
| Overview | Check-in item Pay fines and respond to holding request. |
| Pre-condition | Check-in item and update the item status. |
| Description | Member request to check-in item. Staff or librarian enters item barcode on check-in window. View check-out details. Add item to the collection. Update item status. Update member status. Pas fine. Respond to holding request. |
| Post condition | If success, update checked-in item's status and add entry to the check-in table. If pop up an error message librarian or staff should get required action. |

Table B6: Use case description for check-in item

Use Case Diagram for Generate Reports

Generate reports will be set to the administrator. Librarian and staff can view the generated reports.

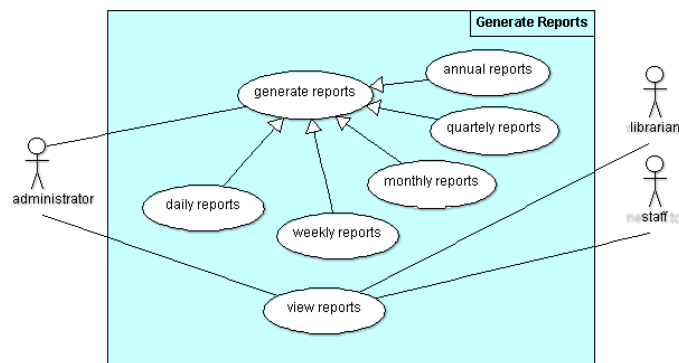


Figure B7: Generate reports module

| | |
|----------------|---|
| Use case name | Generate reports |
| Actors | Administrator, Librarian, Staff |
| Overview | Generate reports. |
| Pre-condition | Use to get feed backs. |
| Description | Generate reports daily, weekly, monthly, quarterly and annually using various criteria. |
| Post condition | Can evaluate the members reading progress. Librarian is known about their students reading progress. |

Table B7: Use case description for report generation

Sequence Diagrams

Sequence Diagram for Member Login

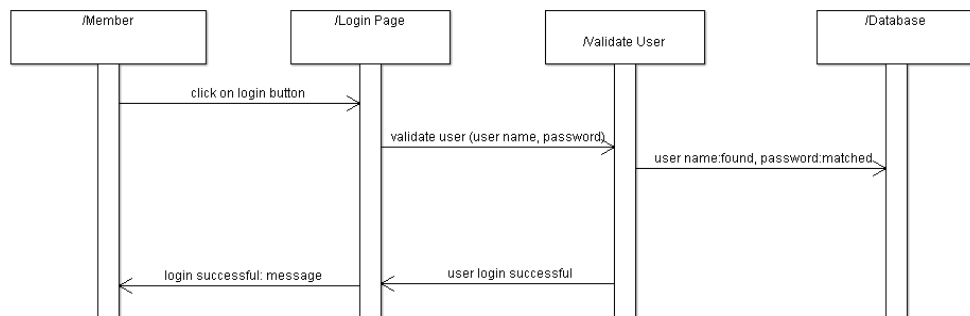


Figure B8: Sequence diagram for member login

Sequence Diagram for Check-in with Fine

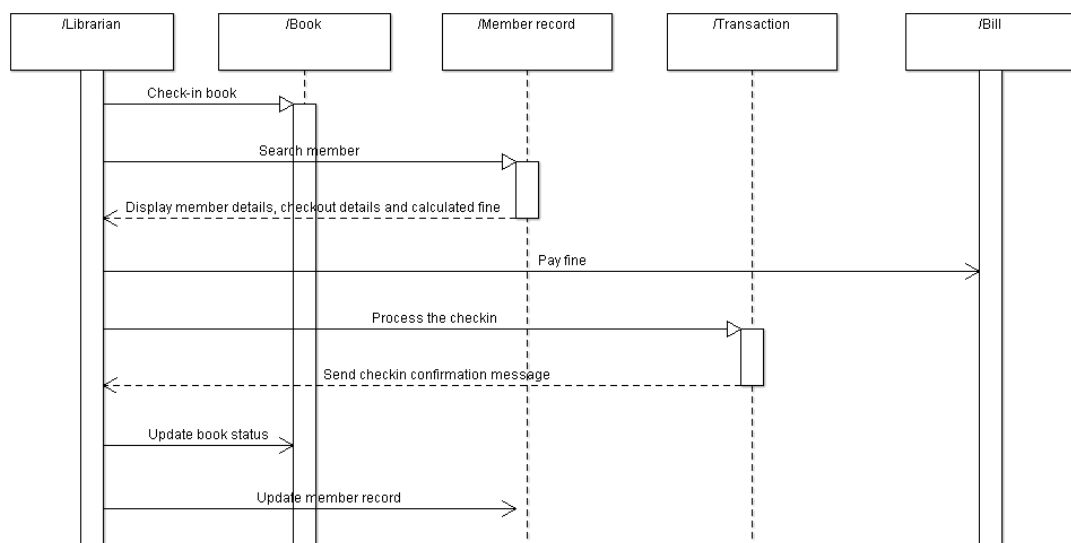


Figure B9: Sequence diagram for check-in with fine

Sequence Diagram for Item Management

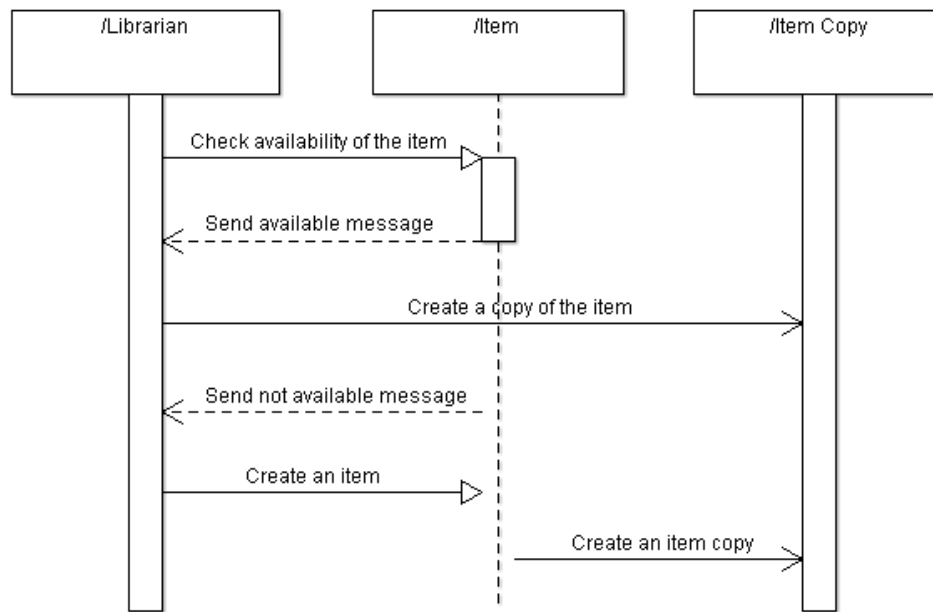


Figure B10: Sequence diagram for item management

Appendix C - User Documentation

This documentation consists with the overall functionalities of the system. Librarian as the administrator, staff, teachers and students as users, can get the benefits of this system.

As the administrator, the librarian is the authorized person to access the whole system. Teachers and the students have granted options. They cannot access the other member details and can't handle circulation activities.

This will be very much effective for the librarian and the staff, because there are less paper work and interestingly work with the system.

Login page

Type the URL http://localhost/VID_SLMS and navigate to the system. First page shows member login form which allows members to log into the system. All levels of members can log into the system in one form. When member tries to log in, the system checks whether this member is valid member or not, otherwise system displays error message. When successfully login in to the system, it directs to the home page of the system according to the member type. Following Figure C.1 shows user login form.

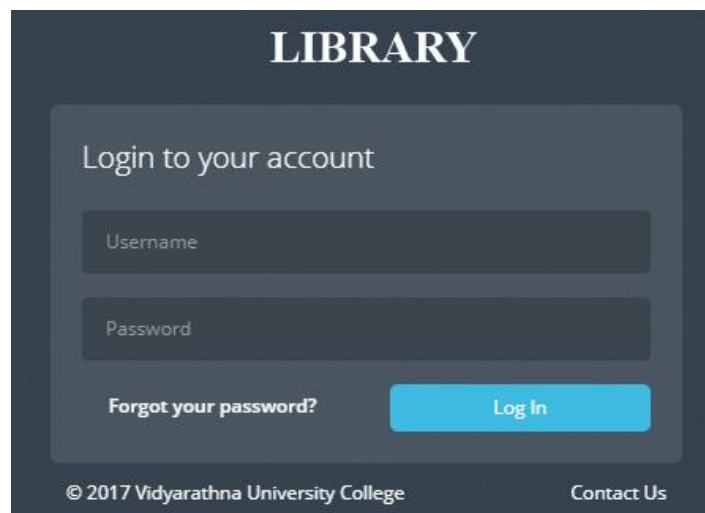


Figure C1: Login page of the system

Dashboard

In the home page, there are some functions related to the library which displays in the dashboard as in figure C2.

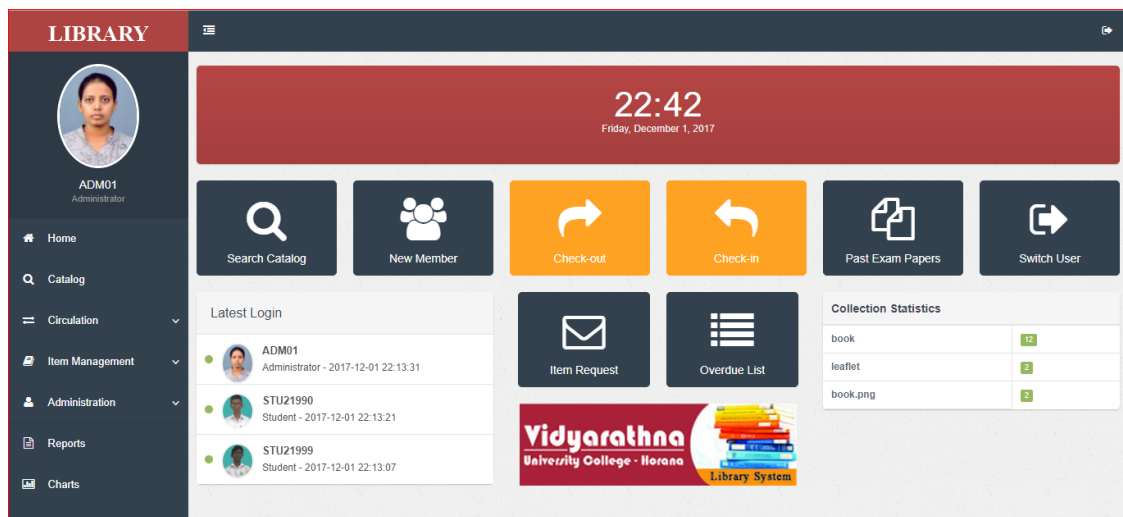


Figure C2: Navigation panel and the dashboard

Logout from the system

To logout from the system, the link can be found on the top right hand corner as in the figure C3. After clicking this link user can successfully logout from the system by selecting “yes” as in the figure C4 and it will direct user back to the Login Page.

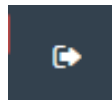


Figure C3: Logout button

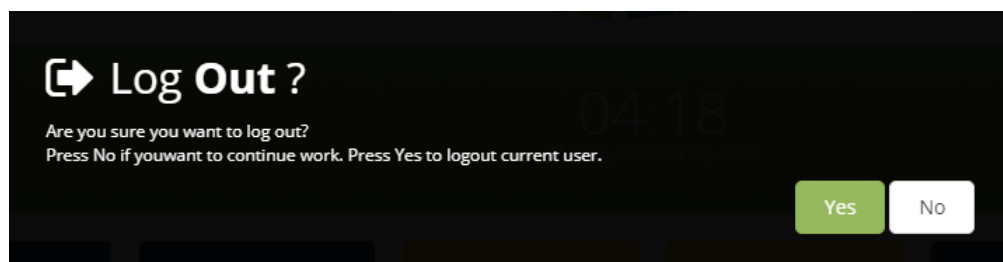


Figure C4: Logout page

Catalog Interface

Catalog interface shows in the figure C5 which used to search items in the collection.

Library Catalog

Book Title Search

Figure C5: Library Catalog page

Circulation Interface

In the circulation interface there are three sub categories as in figure C6.

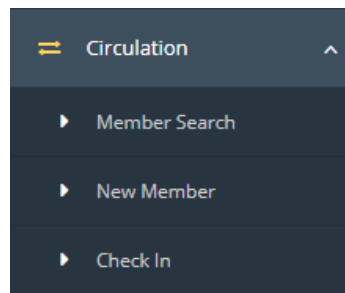


Figure C6: Circulation navigation

By selecting Circulation, admin can navigate to the “Member Search” interface. Member search form attached in figure C7.

Member Search

Search Member by "Membership Number"

Card Number Search

Search Member by "Name"

Name Search

Figure C7: Member search form

1 → If you wish to search member by using “Membership Number”, type the “Membership Number” or “Card Number” here.

2 → Search button

3 → If you wish to search member by using “Member name”, type the name here.

3 → Search button

Admin can also search member by using “Search Member by ‘Name’” panel. It shows member information with equal names. Admin can select the searched member by using review member information and clicking on the member id. Then admin can navigate user

information with check-out and holdings, according to the selected member id. “Member information with equal names” form attached in figure C8.

| Member Information | | | | | | | |
|--------------------|---------|-------|-----------------|--|----------------|-----------------|--------------------|
| Member ID | Picture | Grade | Name | Address | Contact Number | Guardian Mobile | Email |
| STU21990 | 2.png | 10-A | Dineth Sathsara | No. 37/01, Dewana Adiyara, Pinnakola Hena, Walpita, Horana | 0342261536 | 0775675041 | dinethsa@gmail.com |

Figure C8: Member information with equal name form

By using “Search Member by ‘Membership Number’” panel, admin can navigate to the user information with check-out and holdings, according to the entered membership number. “Member information with details” form attached in figure C9.

Member Search

Search Member by "Membership Number"

Search Member by "Name"

Member Information

ADM01

Grade: Other

Name: Ruvini Dilrukshi

Address: No. 343/1, Main Street, Ingiriya

Contact Number: 0342268224

Guardian Mobile: 0714977086

Email: ruvininidsb.it@gmail.com

Expire Date: 2023-12-31

Item Checkout

Item Hold

Item Currently Checked Out

| Checked out date | Category | Barcode | Title | Author | Due Back | Renewal | Days late | Action |
|---------------------|----------|---------|---------------------|----------------------|------------|---------|-----------|----------------------------------|
| 2017-10-18 11:10:34 | 123.jpg | 100 | එයා හරි ඉපුරු ලමයක් | Welikannage Piyadasa | 2017-11-08 | | | <input type="button" value="✕"/> |

Item Currently on Hold

| Place on hold date | Material | Barcode | Title | Author | Status | Due back date | Action |
|---------------------|----------|---------|---------------------|----------------------|--------|---------------|----------------------------------|
| 2017-10-18 23:19:26 | 123.jpg | 104 | එයා හරි ඉපුරු ලමයක් | Welikannage Piyadasa | | | <input type="button" value="✕"/> |

Figure C9: Member information with details form

1 → If you search member by using “Membership Number” or “Name”, It will display the member information here.

2 → If you wish to check-out an item, enter item barcode here.

3 → “Check Out” button

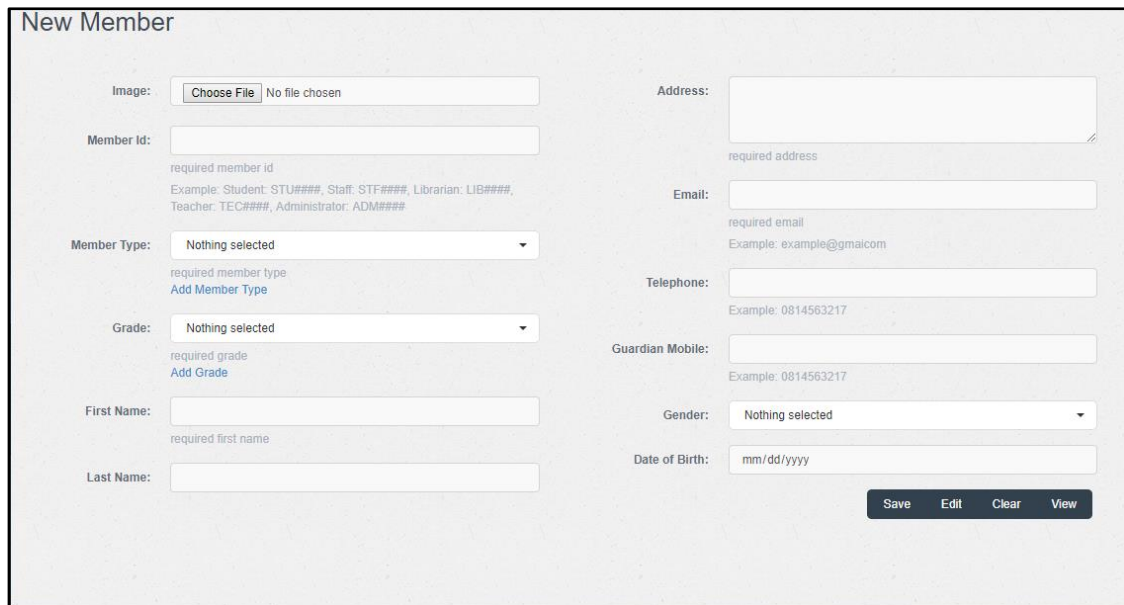
4 → If you wish to hold an item, enter item barcode here.

5 → “Place Hold” button

6 → Shows current check-out of selected member. If you precede check-out an item, the check-out record also adds to this data table.

7 → Shows current hold of selected member. If you precede hold an item, the hold record also adds to this data table.

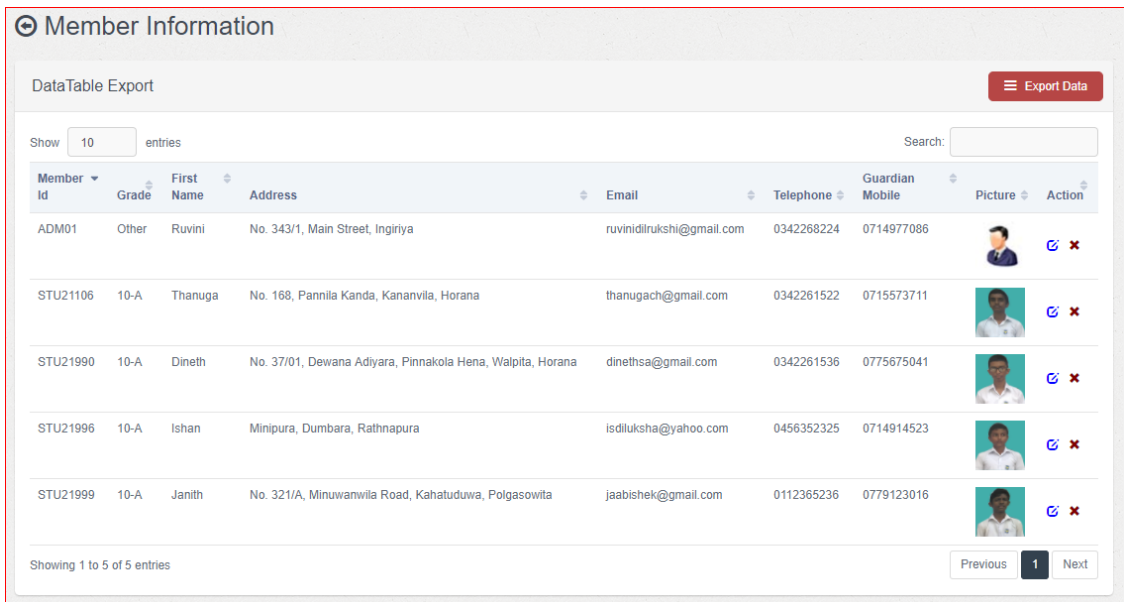
By selecting “New Member” under “Circulation” menu, admin can navigate to the “New Member” form, which can directly register new member. By giving the details of a new member, admin can join a new member to the system. “New Member” form attached in figure C10.



The "New Member" form is a web interface for adding new members. It contains several input fields and dropdown menus. On the left side, there is an "Image" field with a "Choose File" button and "No file chosen" text. Below it is a "Member Id" field with a "required member id" label and an example: "Student: STU####, Staff: STF####, Librarian: LIB####, Teacher: TEC####, Administrator: ADM####". Next is a "Member Type" dropdown menu with "Nothing selected" and a "required member type" label, with a link "Add Member Type". Below that is a "Grade" dropdown menu with "Nothing selected" and a "required grade" label, with a link "Add Grade". At the bottom left are "First Name" and "Last Name" fields with a "required first name" label. On the right side, there is an "Address" field with a "required address" label. Below it is an "Email" field with a "required email" label and an example: "example@gmail.com". Next is a "Telephone" field with an example: "0814563217". Below that is a "Guardian Mobile" field with an example: "0814563217". At the bottom right are a "Gender" dropdown menu with "Nothing selected" and a "Date of Birth" field with a "mm/dd/yyyy" format. At the bottom right corner, there are four buttons: "Save", "Edit", "Clear", and "View".

Figure C10: New member form

After saving the details, admin can view the member information as in following figure C11.



The "Member Information" table is a web interface showing a list of members. It has a "DataTable Export" header with an "Export Data" button. Below the header, there is a "Show" dropdown menu set to "10" and a "Search" input field. The table has the following columns: "Member Id", "Grade", "First Name", "Address", "Email", "Telephone", "Guardian Mobile", "Picture", and "Action". The table contains 5 entries. At the bottom, there is a "Showing 1 to 5 of 5 entries" label and a pagination bar with "Previous", "1", and "Next" buttons.
















| Member Id | Grade | First Name | Address | Email | Telephone | Guardian Mobile | Picture | Action |
|-----------|-------|------------|--|---------------------------|------------|-----------------|---|---|
| ADM01 | Other | Ruvini | No. 343/1, Main Street, Ingiriya | ruvinidilrukshi@gmail.com | 0342268224 | 0714977086 |  |   |
| STU21106 | 10-A | Thanuga | No. 168, Pannila Kanda, Kananvila, Horana | thanugach@gmail.com | 0342261522 | 0715573711 |  |   |
| STU21990 | 10-A | Dineth | No. 37/01, Dewana Adiyara, Pinnakola Hena, Walpita, Horana | dinethsa@gmail.com | 0342261536 | 0775675041 |  |   |
| STU21996 | 10-A | Ishan | Minipura, Dumbara, Rathnapura | isdiluksha@yahoo.com | 0456352325 | 0714914523 |  |   |
| STU21999 | 10-A | Janith | No. 321/A, Minuwanwila Road, Kahatuduwa, Polgasowita | jaabishek@gmail.com | 0112365236 | 0779123016 |  |   |

Figure C11: Member information form



The "Show entries" box is a small web interface with a "Show" label, a dropdown menu set to "10", and an "entries" label.

Using “Show entries” box, admin can see 10 entries at once.

Search:

Using “Search” box, it is helpful to search data record quickly from a data table.

Action

There are two action buttons. First one is for “Edit” and the second one is for “Delete”. Using the “Edit” action button, admin can navigate to the “New Member” form to get it update mode and using “Delete” action button, admin can delete the selected data record.

Export Data

CSV

TXT

XLS

Word

PNG

PDF

Using “Export Data” button, admin can export data on table by selecting appropriate file type. Figure C12 shows exported data of user information by selecting XLS file type.

download (10).xls - Microsoft Excel

| | A | B | C | D | E | F | G | H | I | J |
|---|-----------|-------|------------|--|---------------------------|-----------|-----------------|---------|--------|---|
| 1 | Member Id | Grade | First Name | Address | Email | Telephone | Guardian Mobile | Picture | Action | |
| 2 | ADM01 | Other | Ruvini | No. 343/1, Main Street, Ingiriya | ruvinidilrukshi@gmail.com | 342268224 | 714977086 | | | |
| 3 | STU21106 | 10-A | Thanuga | No. 168, Pannila Kanda, Kananvila, Horana | thanugach@gmail.com | 342261522 | 715573711 | | | |
| 4 | STU21990 | 10-A | Dineth | No. 37/01, Dewana Adiyara, Pinnakola Hena, Walpita, Horana | dinethsa@gmail.com | 342261536 | 775675041 | | | |
| 5 | STU21996 | 10-A | Ishan | Minipura, Dumbara, Rathnapura | isdiluksha@yahoo.com | 456352325 | 714914523 | | | |
| 6 | STU21999 | 10-A | Janith | No. 321/A, Minuwanwila Road, Kahatuduwa, Polgasowita | jaabishek@gmail.com | 112365236 | 779123016 | | | |
| 7 | | | | | | | | | | |
| 8 | | | | | | | | | | |
| 9 | | | | | | | | | | |

Figure C12: Exported data of user information

By selecting Circulation, admin can navigate to the “Check In” interface. Using that interface admin can check in items by entering item barcode number and click “Add to Collection” button. “Check In” form attached in figure C13.

Check In

Item Check In

Barcode Number Search

| | |
|----------------|---------------------|
| Member ID | STU23601 |
| Name | Vimukthi Kavinda |
| Grade | 6-A |
| Barcode | 102 |
| Title | ඉරට හඳට මම නොහඬමි |
| Author | Pathirana U. P. S. |
| Check-out date | 2017-11-02 00:26:51 |
| Check-in date | 2017-11-16 00:00:00 |
| Fine | |

Check In

Figure C13: Check in form

1 → If you wish to check-in an item, enter item “Barcode Number” here.

2 → Search button.

3 → Display the item and member information which related to the entered barcode.

4 → “Check In” button. If you click that button, it will update the item status (available) and disable the check-out record which related to the member id.

Item Management Navigation

In the “Item Management” interface there are two sub categories as in figure C14.

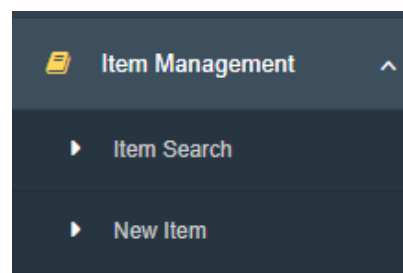


Figure C14: Circulation navigation

By selecting “Item Management”, admin can navigate to the “Item Search” interface which shows in figure C15, that interface helps to create an item copy. It consist “Search Item by ‘ISBN Number’” panel and “Search Item by ‘Search Phrase’” panel. If admin searches item by ISBN number, the system will display the information about item and information about copies of the item. The “Create Item Copy” button in the “Item Information” panel helps to create a copy of the item.

“Search Item by ‘Search Phrase’” panel helps to search item by using title, author, publisher, subject, collection or item call number.

Item Search

Search Item by "ISBN Number"
 Search

Search Item by "Search Phrase"
 Search

Item Information

ITM00004 [Create Item Copy](#)

| | |
|-------------|--|
| Title | ඉරට් හඳව් මම නොහඳව්මි |
| Author | Pathirana U. P. S. |
| Publisher | Sudarshana Printing & Publishing, Karadana |
| ISBN | 9789551611019 |
| Subject | Sinhala novel |
| Collection | lending |
| Call Number | |

Item Copy Information

| Barcode | Location | Notes | Acquisition Date |
|---------|----------|-------|---------------------|
| 102 | Rack_1 | | 2017-10-24 10:56:16 |
| 09 | Rack_2 | | 2017-10-27 04:34:36 |

Figure C15: Item search form

1 → If you wish to search item by using ISBN number, enter ISBN number here.

2 → Search button

3 → If you wish to search item by using search phrase (title, author, publisher, subject, collection, call number), enter search phrase here.

4 → Search button

5 → If you search item by using ISBN number or search phrase, it will display the item information here.

6 → “Create Item Copy” button. If you click “Create Item Copy” button, it will facilitate to create an item copy which related to the selected item.

7 → Display item copy information, which relate to the selected item.

Administration Navigation

All the other modules which use to handle main process are shown under “Administration” menu. It consists 8 sub modules. (Figure C16)

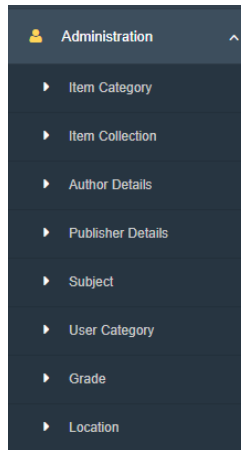


Figure C16: Administration navigation

Reports Interface

Reports interface helps to get many reports which are useful to measure the performances of the library and members. Figure C17 shows reports interface.

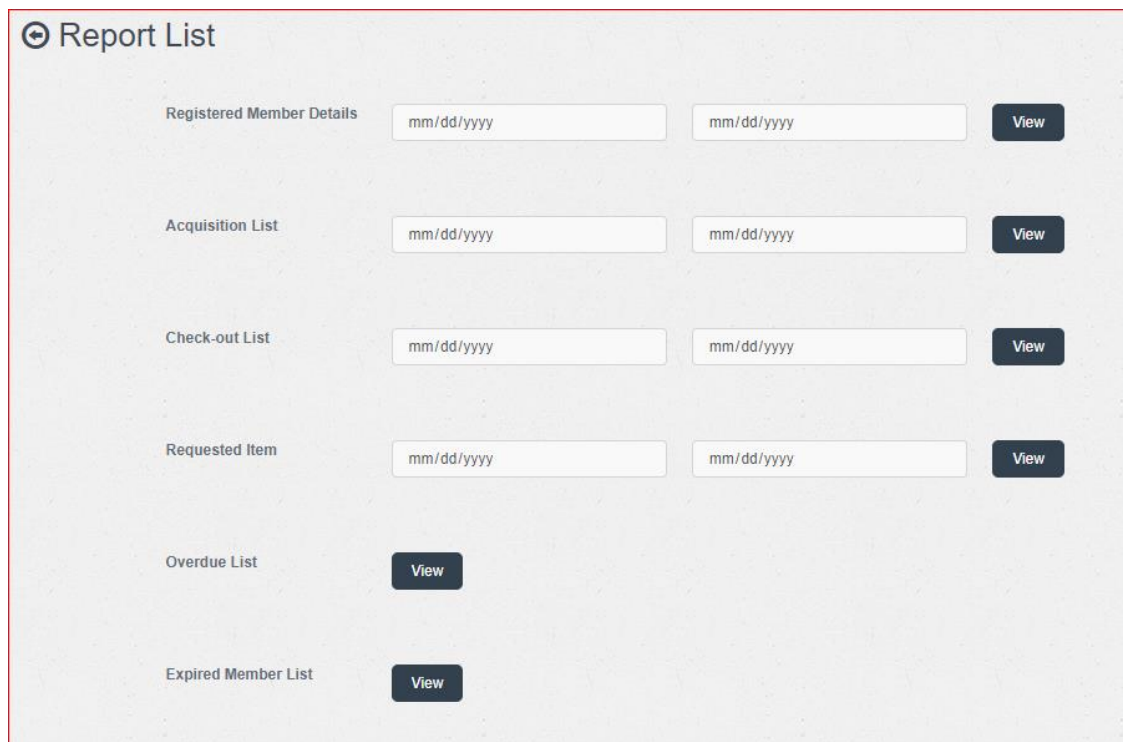


Figure C17: Reports interface

Charts Interface

According to the details of the function, charts can be generated. Figure C18 shows the chart interface. C19 shows the bar chart of login types and C20 shows pie chart of login types of the users of the system.

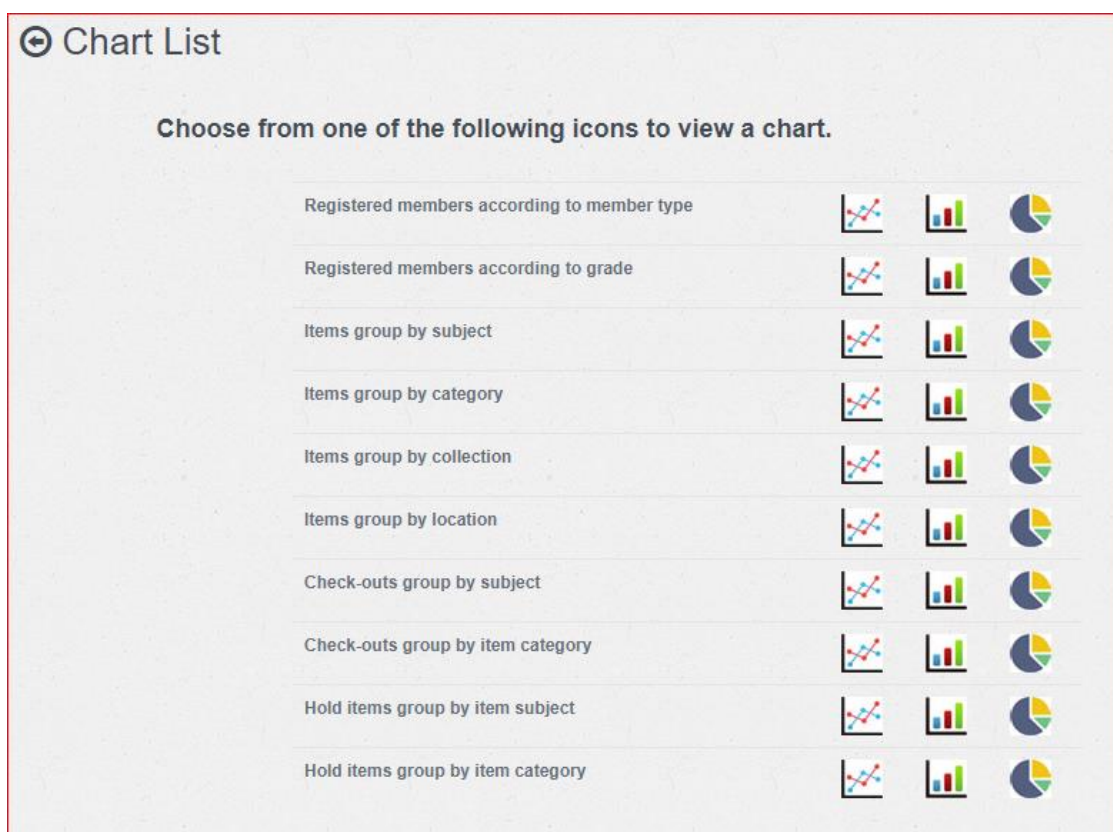


Figure C18: Charts interface

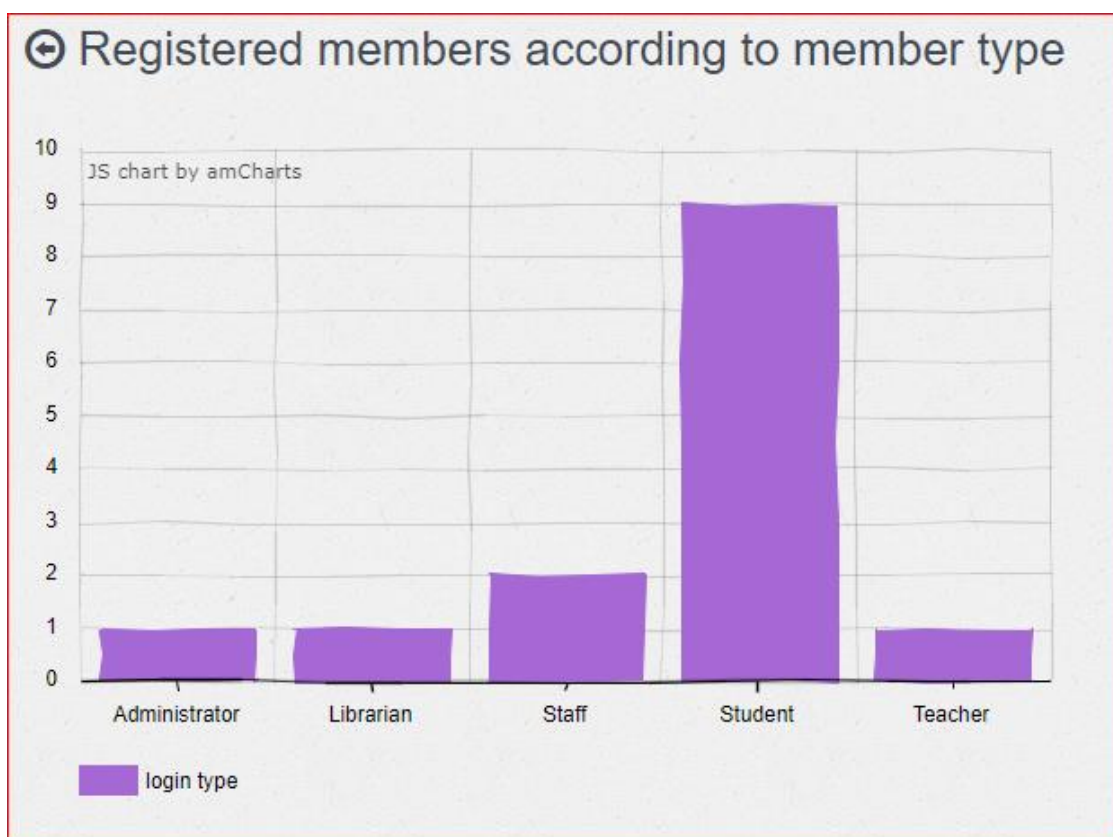


Figure C19: Bar chart of login types

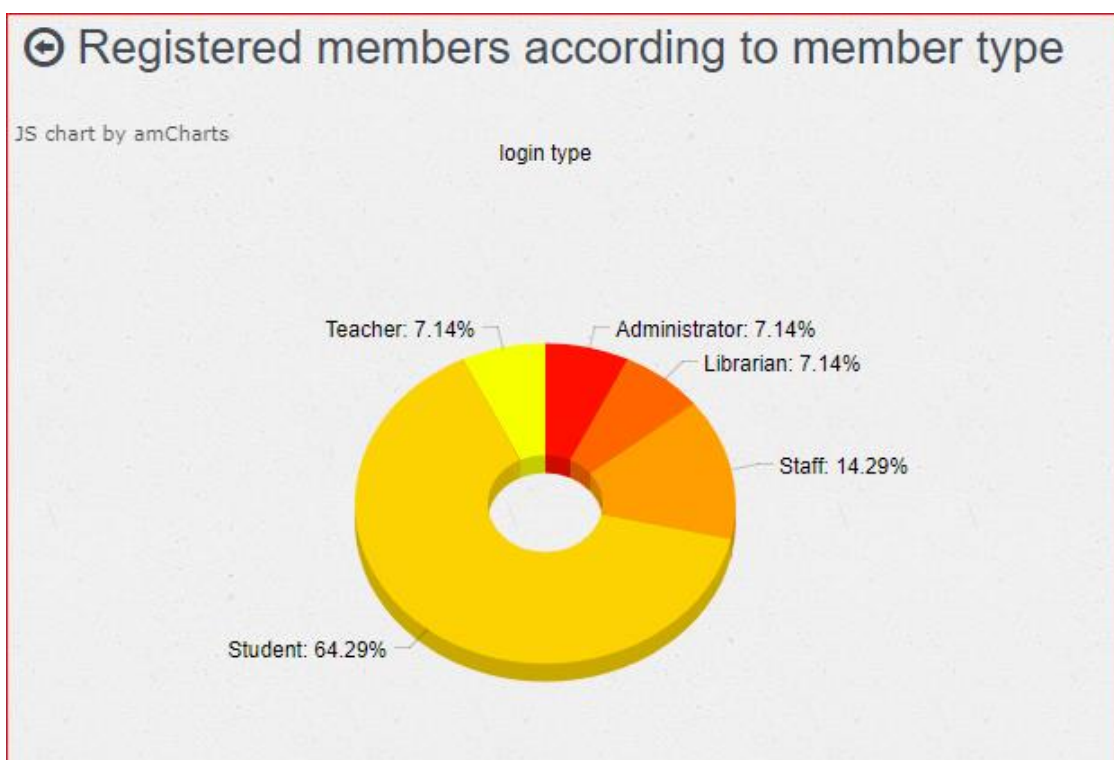


Figure C20: Pie chart of login types

Appendix D – Management Reports

Reports can be taken as the summary reports based on the library and member performance. Some of the generated reports are shown below.

Login Type Report

Figure D1 shows login type report.

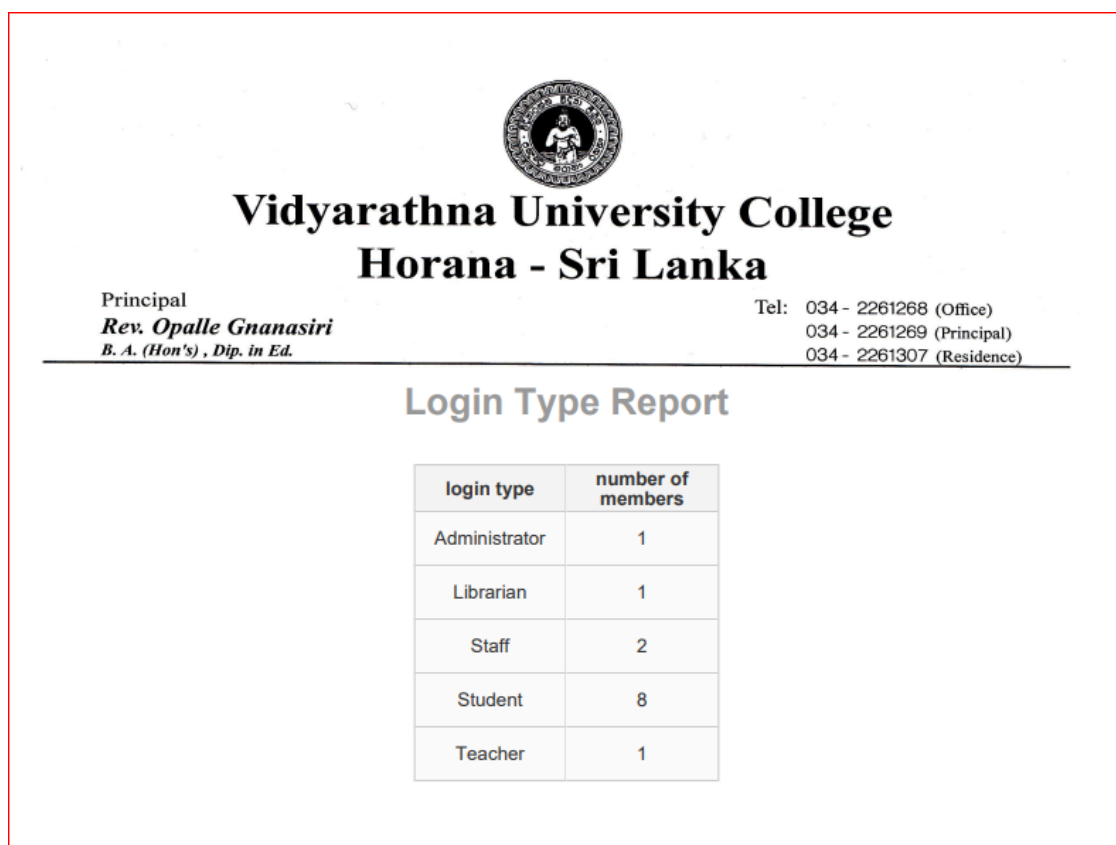


Figure D1: Login type report

Member Detail Report

Figure D2 shows member detail report.



Vidyarathna University College Horana - Sri Lanka

Principal
Rev. Opalle Gnanasiri
B. A. (Hon's) , Dip. in Ed.

Tel: 034 - 2261268 (Office)
034 - 2261269 (Principal)
034 - 2261307 (Residence)

Member Detail Report

| Member ID | Grade | Name | Address | Email | Contact |
|-----------|-------|---------------------|--|--------------------|------------|
| ADM01 | Other | Ruvini Dilrukshi | No. 343/1, Main Street, Ingiriya | ruvinidilrukshi@gr | 0342268224 |
| LIB01 | Other | Lakshman Kumara | No. 23 B, Main Street, Horana | lakikuma@gmail.c | 0342261572 |
| STF01 | Other | Sarath Ekanayake | No. 123, Padukka Road, Horana | eksarath@gmail.c | 0342263852 |
| STF02 | Other | Janaki Fernando | No. 42, Idama, Moratuwa | hnjmsd@gmail.co | 0112635289 |
| STU21106 | 10-A | Thanuga Chantuka | No. 168, Pannila Kanda, Kananvila, Horana | thanugach@gmail | 0342261522 |
| STU21990 | 10-A | Dineth Sathsara | No. 37/01, Dewana Adiyara, Pinnakola Hena, Walpita, Horana | dinethsa@gmail.c | 0342261536 |
| STU21996 | 10-A | Ishan Diluksha | Minipura, Dumbura, Rathnapura | isdiluksha@yahoo | 0456352325 |
| STU21999 | 10-A | Janith Abhishek | No. 321/A, Minuwanwila Road, Kahatuduwa, Polgasowita | jaabishek@gmail. | 0112365236 |
| STU22001 | 10-A | Madura Dannanjaya | Minikapugodella, Diyaparella, Govinna | madura@yahoo.co | 0342563528 |
| STU22008 | 10-A | Amila Yasmitha | 47/C, Tissa Mawatha, Pamunugama, Alubomulla | yasmithaa@gmail | 0382653628 |
| STU22013 | 10-A | Sithum Udaraka | No. 53, Aluth Hena, Maha Uduwa, Kuda Uduwa | siudaraka@gmail. | 0342265879 |
| STU22026 | 10-A | Yasith Ruwantha | 181/B, Mavil Ela Road, Aluthgama, Bandaragama | yrwantha@yahoo | 0342256856 |
| TEC01 | Other | Ruwani Jayawardhana | "Ramya", Panadura Road, Horana | ruwanijay@gmail. | 0342256368 |

Figure D2: Member detail report

Check-out Report

Figure D3 shows check-out report.



Vidyarathna University College Horana - Sri Lanka

Principal
Rev. Opalle Gnanasiri
B. A. (Hon's) , Dip. in Ed.

Tel: 034 - 2261268 (Office)
034 - 2261269 (Principal)
034 - 2261307 (Residence)

Check-out Report

| Member ID | Item Barcode | Check-out Date | Check-in Date |
|-----------|--------------|------------------------|------------------------|
| ADM01 | 101 | 2017-10-30 22:21:14 | 2017-11-13 00:00:00 |
| STF02 | 105 | 2017-11-01 23:56:55 | 2017-11-15 00:00:00 |
| STU150 | 103 | 2017-11-01 23:35:22 | 2017-11-15 00:00:00 |
| STU23601 | 102 | 2017-11-02 00:26:51 | 2017-11-16 00:00:00 |
| STU23601 | 100 | 2017-11-02 00:31:32 | 2017-11-16 00:00:00 |

Figure D3: Check-out report

Appendix E – Test Results

All the test results are shown in this Appendix. The following tests were carried out using major test cases which mentioned in the Evaluation chapter.

Test results for the User Authentication

The test cases with results belong to User Authentication is displayed by Table E1.

| No | Test Description | Steps to Test | Expected Result |
|----|-----------------------------|--|---|
| 1. | Validate user Input details | Enter correct user name and password | Welcome to Vidyaratna Library |
| 2. | Validate user Input details | Enter correct user name and incorrect password | Please check your password! |
| 3. | Validate user Input details | Enter incorrect user name and correct password | Please check your user name and password or invalid user! |
| 4. | Validate user Input details | Enter both user name and password incorrectly | Please check your user name and password or invalid user! |

Table E1: User authentication

Test results for Member Registration Module

The test cases with results belong to Member Registration Module is displayed by table E2.

| No | Test Description | Steps to Test | Expected Result |
|----|-------------------------------|---|---|
| 1. | Add member details | Enter details correctly | Created user "STU568" successfully |
| 2. | Add member details | Save without entering data | ! Please fill out this field. |
| 3. | Add member details | Enter invalid member id | ! Please match the requested format. |
| 4. | Add member details | Enter incorrect email address | ! Please include an '@' in the email address. 'sfsf' is missing an '@'. |
| 5 | Validate member input details | Viewing the entered data of the members | Redirect to the "Member Information" page |
| 6 | Delete member record | Delete selected record | Removed member "STU256" successfully |
| 7 | Edit member record | Viewing the selected member details | Updated user "ADM01" successfully |

Table E2: Member registration

Test results for Check-out

The test cases with results belong to Check-out Module is displayed by table E3.

| No | Test Description | Steps to Test | Expected Result |
|----|----------------------------|---|-------------------------------|
| 1 | Search member for checkout | Enter correct card number | Display member information |
| 2 | Search member for checkout | Enter incorrect card number | Enter correct card number |
| 3 | Validate member | Enter correct barcode. If the member is not exceeding the check-out limit, checked-out the selected item. | Check-out item successfully |
| 7 | Validate member | Enter correct barcode. If the member is exceeding check-out limit. | Exceed your check-out limit |
| 8 | Validate member | Enter correct barcode. If the member with fine. | Pay fine |
| 9 | Validate member | Enter correct barcode. If the member is in-active. | User expired. can't check-out |
| 10 | Validate entered barcode | Enter correct barcode. Check availability of entered barcode. If it is not available. | Item not available |
| 11 | Validate entered barcode | Enter incorrect barcode. | Enter correct barcode |

Table E3: Check-out

Test results for Check-in

The test cases with results belong to Check-in Module is displayed by table E4.

| No | Test Description | Steps to Test | Expected Result |
|----|---|---|---|
| 1 | Search check-out record | Enter item barcode | Display check-out details, member information and fine if any |
| 2 | Search check-out record | Enter invalid item barcode | Not a valid barcode |
| 3 | Verify member details, and pay fine, if there is. | Verify member details and click "Pay Fine" button | Paid |
| 4 | Verify member details, release check-out record and change the status of the item | Verify member details and click "Check In" button | Item successfully check-in |

Table E4: Check-in

Appendix F - Code Listing

Some of the code segments of the system are attached to the Appendix F.

Login Interface Cording

Coding segment of the login interface page is as follows.

```
<?php
require_once 'apps/control/config.php';
$template = new template();
?>

<!DOCTYPE html>
<html lang="en" class="body-full-height">
  <?php $template->getHead(); ?>
  <body>

    <!-- START LOGIN CONTAINER -->
    <div class="login-container">

      <div class="login-box animated fadeInDown">

        <!-- LIBRARY LOGO -->
        <div class="login-logo"></div>
        <!-- LIBRARY LOGO -->

        <div class="login-body">

          <!-- DISPLAY SUCCESS AND ERROR MESSAGE -->
          <?php $template->getMsg(); ?>
          <!-- DISPLAY SUCCESS AND ERROR MESSAGE -->

          <div class="login-title">Login to your account</div>

          <!-- START LOGIN FORM -->
          <form class="form-horizontal" method="post" action='apps/model/UserRegMgr/_login.php'>

            <div class="form-group">
              <div class="col-md-12">
                <input type="text" class="form-control" placeholder="Username" name="login_userName"/>
              </div>
            </div>

            <div class="form-group">
              <div class="col-md-12">
                <input type="password" class="form-control" placeholder="Password" name="login_password"/>
              </div>
            </div>

            <div class="form-group">
              <div class="col-md-6">
                <a href="#" class="btn btn-link btn-block">Forgot your password?</a>
              </div>
              <div class="col-md-6">
                <input type="SUBMIT" name="LOGIN" value="Log In" class="btn btn-info btn-block">
              </div>
            </div>

          </form>
          <!-- END LOGIN FORM -->

        </div>
        <div class="login-footer">
          <div class="pull-left">
            &copy; 2017 Vidyarthna University College
          </div>
          <div class="pull-right">
            <a href="#">Contact Us</a>
          </div>
        </div>
      </div>
    </div>
    <!-- END LOGIN CONTAINER -->

  </body>
</html>
```

Login Handling Coding

Coding segment of the login handling page is as follows.

```
<?php
require_once '../control/config.php';

if (isset($_POST['LOGIN'])) {

    $login_userName = $_POST['login_userName']; // user name as library registration number
    $login_password = sha1($_POST['login_password']); // encrypted password

    try {
        // select active member
        $sql = "SELECT login_userName, login_password, login_type, user_pic, user_expDate FROM tbl_login, tbl_user
        WHERE login_userName=user_memberId and user_status='Active'
        and login_userName=:login_userName and user_expDate > CURDATE()";

        $stmt = $conn->prepare($sql);
        $stmt->execute(array(':login_userName' => $login_userName));
        $result = $stmt->fetchAll();

        if (count($result)) {
            $row = $result[0];

            // assign DB password to variable
            $login_passwordDB = $row['login_password'];

            // check DB password with user entered password
            if ($login_passwordDB == $login_password) {

                // assign username and login type to session
                $_SESSION['username'] = $row['login_userName'];
                $_SESSION['login_type'] = $row['login_type'];
                $_SESSION['user_pic'] = $row['user_pic'];

                // success message & redirect to home page
                $_SESSION['SUCESSS'] = 'Welcome to Vidyarathna Library ';

                if ($_SESSION['login_type'] == 'Administrator') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeAdm.php');
                }
                if ($_SESSION['login_type'] == 'Librarian') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeLibrarian.php');
                }
                if ($_SESSION['login_type'] == 'Staff') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeStaff.php');
                }
                if ($_SESSION['login_type'] == 'Teacher') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeOther.php');
                }
                if ($_SESSION['login_type'] == 'Student') {
                    header("Location: " . $_SERVER['BASE_URL'] . '/apps/view/Home/HomeOther.php');
                }
            } else {

                // display if password is incorrect
                $_SESSION['ERROR'] = 'Please check your password!';
                header("Location: " . $_SERVER['HTTP_REFERER']);
            }

            // display incorrect user or inactive user
            $_SESSION['ERROR'] = 'Please check your user name and password or invalid user!';
            header("Location: " . $_SERVER['HTTP_REFERER']);
        }
    } catch (Exception $ex) {
        $_SESSION['ERROR'] = $ex->getMessage();
        header("Location: " . $_SERVER['HTTP_REFERER']);
    }
}

?>
```

Log Out Cording

Coding segment of the log out is as follows. This will redirect to the index page (login page) of the system.

```
<?php
require_once '../control/config.php';

session_destroy();

header("Location: " . $_SERVER['BASE_URL']);
```

Member Registration Coding

Coding segment of the member registration is as follows.

```
<?php
require_once '../control/config.php';
$template = new template();

// check user logged and user role to access page
if ((isset($_SESSION['username']) && ($_SESSION['login_type'] == 'Administrator')) ||
    (isset($_SESSION['username']) && ($_SESSION['login_type'] == 'Librarian')) ||
    (isset($_SESSION['username']) && ($_SESSION['login_type'] == 'Staff')) {

    // allow to access pg if isset
    if (isset($_GET['user_memberId'])) {

        // get member id from url and assign to variable
        $user_memberId = $_GET['user_memberId'];

        // query the particular user detail from the given user id
        $sql = 'SELECT user_grade, user_fname, user_lname, user_address, user_email, user_tel, user_guardianMob, '
            . 'user_gender, user_dob, user_pic, login_type FROM db_slms.tbl_user, db_slms.tbl_login '
            . 'WHERE user_memberId=login_userName AND user_memberId=:user_memberId';

        $stmt = $conn->prepare($sql);
        $stmt->execute(array('user_memberId' => $user_memberId));

        while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
            // assign the values to variables before pass to DB
            $user_grade = $row[0];
            $user_fname = $row[1];
            $user_lname = $row[2];
            $user_address = $row[3];
            $user_email = $row[4];
            $user_tel = $row[5];
            $user_guardianMob = $row[6];
            $user_gender = $row[7];
            $user_dob = $row[8];
            $user_pic = $row[9];
            $login_type = $row[10];
        }
    }
} else {
    //not redirect to login pg
    header("Location: " . $_SERVER['BASE_URL'] . 'index.php');
}

?>
<!DOCTYPE html>
<html lang="en">

    <?php $template->getHead(); ?>

    <body>

        <!-- START PAGE CONTAINER -->
        <div class="page-container">

            <!-- START PAGE SIDEBAR -->
            <?php $template->getSidebar(); ?>
            <!-- END PAGE SIDEBAR -->

            <!-- PAGE CONTENT -->
            <div class="page-content">

                <!-- START X-NAVIGATION VERTICAL -->
                <?php $template->getNavigation(); ?>
                <!-- END X-NAVIGATION VERTICAL -->

                <!-- PAGE TITLE -->
                <div class="page-title">
                    <h2> New Member</h2>
                </div>
                <!-- END PAGE TITLE -->

                <!--PAGE CONTENT WRAPPER-->
                <div class="page-content-wrap">
                    <div class="row">

                        <!-- this function display success and error message -->
                        <?php $template->getMsg(); ?>
                        <!-- this function display success and error message -->

                        <form enctype="multipart/form-data" id="jvalidate" role="form" method="POST"
                            class="form-horizontal" action="../../model/UserRegMgr/_UserRegister.php">

                            <div class="col-md-6">

                                <!--START JQUERY VALIDATION PLUGIN-->
                                <div class="block">
                                    <div class="panel-body">
```

```

<div class="form-group">
<label class="col-md-3 control-label">Image: </label>
<div class="col-md-9">
<input type="file" class="form-control" name="user_pic" autofocus="" value=<?php
if (isset($_GET['user_memberId'])) {
echo $user_pic;
}
?> />
<span class="help-block"><?php
if (isset($_GET['user_memberId'])) {
echo $user_pic;
}
?> </span>
</div>
</div>

<div class="form-group">
<label class="col-md-3 control-label">Member Id:</label>
<div class="col-md-9">
<input type="text" class="form-control" name="user_memberId" pattern="[A-Z]{3}[0-9]{1,}"
if (isset($_GET['user_memberId'])) {
echo $user_memberId;
}
?> >
<span class="help-block">required member id</span>
<span class="help-block">Example: Student: STU###, Staff: STF###, Librarian: LIB###,
</div>
</div>

<...26 lines />
<...23 lines />
<...11 lines />
<...10 lines />

</div>
</div>
<!--END JQUERY VALIDATION PLUGIN-->
</div>

<...90 lines />
</form>
</div>
</div>
<!--END PAGE CONTENT WRAPPER-->

</div>
<!--END PAGE CONTENT-->

</div>
<!-- END PAGE CONTAINER -->

<!-- MESSAGE BOX-->
<?php $template->getMessage(); ?>
<!-- END MESSAGE BOX-->

<!-- START SCRIPTS -->
<?php $template->getScript(); ?>
<!-- END SCRIPTS -->

</body>
</html>

```

Member Registration Handling Coding

Coding segment of the member registration handling is as follows.

```

<?php
require_once '../control/config.php';

// Assign values passed from Form to variables before insert to DB
$user_memberId = $_POST['user_memberId'];
$user_grade = $_POST['user_grade'];
$user_fname = $_POST['user_fname'];
$user_lname = $_POST['user_lname'];
$user_address = $_POST['user_address'];
$user_email = $_POST['user_email'];
$user_tel = $_POST['user_tel'];
$user_guardianMob = $_POST['user_guardianMob'];
$user_gender = $_POST['user_gender'];
$user_dob = $_POST['user_dob'];

// $user_status = 'Active';
// return first two letters of a grade (grade 1 - 13)
$user_expDate = substr($user_grade, 0, 2);

```

```

// calculate user expire year according to the grade
$yearnow = date("Y");
switch ($user_expDate) {...53 lines }

//concatenate expire 'year' and 'mm-dd'
$user_expDate = $user_expDate . '-12-31';

if ($user_expDate < date("Y-m-d")) {...3 lines } else {
    $user_status = 'Active';
}
$login_userName = $user_memberId;
$login_password = sha1($user_memberId); //encrypt pwd
$login_type = $_POST['user_type'];

// To handle errors
try {

    // upload user picture -----
    $allowedExts = array("gif", "jpeg", "jpg", "png");
    $temp = explode(".", $_FILES["user_pic"]["name"]);
    $extension = end($temp);

    if (true) {...14 lines } else {
        $_SESSION['ERROR'][] = "Invalid file";
    }

    // upload user picture -----

    // if user press SAVE button execute below code and add new record to DB
    if (isset($_POST['Save'])) {

        //===== Create User
        // insert SQL
        $sql = 'INSERT INTO tbl_user(user_memberId,user_grade,user_fname,user_lname,user_address,user_email,user_tel,'
            . 'user_guardianMob,user_gender,user_dob,user_pic,user_expDate,user_status) '
            . 'VALUES(:user_memberId,:user_grade,:user_fname,:user_lname,:user_address,:user_email,:user_tel,'
            . ':user_guardianMob,:user_gender,:user_dob,:user_pic,:user_expDate,:user_status)';

        // connecting to DB
        $stmt = $conn->prepare($sql);

        // pass values to DB
        $stmt->execute(array('user_memberId' => $user_memberId,
            'user_grade' => $user_grade,
            'user_fname' => $user_fname,
            'user_lname' => $user_lname,
            'user_address' => $user_address,
            'user_email' => $user_email,
            'user_tel' => $user_tel,
            'user_guardianMob' => $user_guardianMob,
            'user_gender' => $user_gender,
            'user_dob' => $user_dob,
            'user_pic' => $user_pic,
            'user_expDate' => $user_expDate,
            'user_status' => $user_status));

        //===== Create Login
        // insert SQL
        $sqlL = 'INSERT INTO tbl_login(login_userName,login_password,login_type) '
            . 'VALUES(:login_userName,:login_password,:login_type)';

        // connecting to DB
        $stmtL = $conn->prepare($sqlL);

        // pass values to DB
        $stmtL->execute(array('login_userName' => $login_userName,
            'login_password' => $login_password,
            'login_type' => $login_type));

        // after success disconnect from DB
        $conn = null;

        // print success msg
        $_SESSION['SUCCESS'] = 'Created user ' . "" . $user_memberId . "" . ' successfully';
    }

    // if user press EDIT button (After SAVE the record), execute below code and edit the current record on the DB
    if (isset($_POST['Edit'])) {

        //===== Edit User
        // update SQL
        $sql = 'UPDATE tbl_user '
            . 'SET user_grade=:user_grade, user_fname=:user_fname, user_lname=:user_lname, user_address=:user_address,'
            . 'user_email=:user_email, user_tel=:user_tel, user_guardianMob=:user_guardianMob, user_gender=:user_gende'
            . 'user_dob=:user_dob, user_expDate=:user_expDate, user_status=:user_status '
            . 'WHERE user_memberId=:user_memberId';

        // connecting to DB
        $stmt = $conn->prepare($sql);

```

```

// pass values to DB
$stmt->execute(array('user_memberId' => $user_memberId,
    'user_grade' => $user_grade,
    'user_fname' => $user_fname,
    'user_lname' => $user_lname,
    'user_address' => $user_address,
    'user_email' => $user_email,
    'user_tel' => $user_tel,
    'user_guardianMob' => $user_guardianMob,
    'user_gender' => $user_gender,
    'user_dob' => $user_dob,
    'user_expDate' => $user_expDate,
    'user_status' => $user_status));

//===== Edit Login
// update SQL

$sqlL = 'UPDATE tbl_login SET login_password=:login_password, login_type=:login_type '
    . 'WHERE login_userName=:login_userName';

// connecting to DB
$stmtL = $conn->prepare($sqlL);

// pass values to DB
$stmtL->execute(array('login_userName' => $login_userName,
    'login_password' => $login_password,
    'login_type' => $login_type));

// print success msg
$_SESSION['SUCCESS'] = 'Updated user ' . ' ' . $user_memberId . ' ' . ' successfully';
}

// Print error msg while ...
catch (Exception $ex) {
    // if ($ex->getCode() == '23000') {
    //     $_SESSION['ERROR'] = ' User Member Id ' . ' ' . $user_memberId . ' ' . ' already exists ... try another !!!';
    // } else {
    $_SESSION['ERROR'] = $ex->getMessage();
    echo $ex->getMessage();
    // }
}

//redirect from module page to view page
header("Location: " . $_SERVER['HTTP_REFERER']);
??

```

Member Registration View Coding

After entering the details of the new member to the database, admin can view the list of member by using this page. Coding segment of the view of the member registration is as follows.

```

<?php
require_once '../control/config.php';
$template = new template();
??

<!DOCTYPE html>
<html lang="en">

    <?php $template->getHead(); ??>

    <body>
        <!-- START PAGE CONTAINER -->
        <div class="page-container">

            <!-- START PAGE SIDEBAR -->
            <?php $template->getSidebar(); ??>
            <!-- END PAGE SIDEBAR -->

            <!-- PAGE CONTENT -->
            <div class="page-content">

                <!-- START X-NAVIGATION VERTICAL -->
                <?php $template->getNavigation(); ??>
                <!-- END X-NAVIGATION VERTICAL -->

                <!-- PAGE TITLE -->
                <div class="page-title">
                    <h2><span class="fa fa-arrow-circle-o-left"></span> Member Information</h2>
                </div>
            </div>
        </div>
    </body>
</html>

```

```

<!-- START DEFAULT DATATABLE -->
<div class="panel panel-default">

    <!-- DATA EXPORT PANEL -->
    <div class="panel-heading">
        <h3 class="panel-title">DataTable Export</h3>
        <div class="btn-group pull-right">
            <button class="btn btn-danger dropdown-toggle" data-toggle="dropdown"><i class="fa fa-bars"></i>
                Export Data</button>
            <ul class="dropdown-menu">
                <li class="divider"></li>
                <li><a href="#" onClick = '$("#member").tableExport({type:"csv",escape:"false"});'>
                    <img src='../.../lib/img/icons/csv.png' width="24"/> CSV</a></li>
                <li><a href="#" onClick = '$("#member").tableExport({type:"txt",escape:"false"});'>
                    <img src='../.../lib/img/icons/txt.png' width="24"/> TXT</a></li>
                <li class="divider"></li>
                <li><a href="#" onClick = '$("#member").tableExport({type:"excel",escape:"false"});'>
                    <img src='../.../lib/img/icons/xls.png' width="24"/> XLS</a></li>
                <li><a href="#" onClick = '$("#member").tableExport({type:"doc",escape:"false"});'>
                    <img src='../.../lib/img/icons/word.png' width="24"/> Word</a></li>
                <li class="divider"></li>
                <li><a href="#" onClick = '$("#member").tableExport({type:"png",escape:"false"});'>
                    <img src='../.../lib/img/icons/png.png' width="24"/> PNG</a></li>
                <li><a href="#" onClick = '$("#member").tableExport({type:"pdf",escape:"false"});'>
                    <img src='../.../lib/img/icons/pdf.png' width="24"/> PDF</a></li>
            </ul>
        </div>
    </div>
    <!-- END DATA EXPORT PANEL -->

    <div class="panel-body">
        <table id="member" class="table datatable">
            <thead>
                <tr>
                    <th>Member Id</th>
                    <th>Grade</th>
                    <th>First Name</th>
                    <th>Address</th>
                    <th>Email</th>
                    <th>Telephone</th>
                    <th>Guardian Mobile</th>
                    <th>Picture</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody>
                <?php
                //connect to DB and get all the user details
                $sql = 'SELECT * FROM db_slms.tbl_user';
                $stmt = $conn->prepare($sql);
                $stmt->execute(array());

                while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
                    // execute each row and display data in a row
                    ?>

                    <tr>
                        <td><?php echo $row[1]; ?></td>
                        <td><?php echo $row[2]; ?></td>
                        <td><?php echo $row[3]; ?></td>
                        <td><?php echo $row[5]; ?></td>
                        <td><?php echo $row[6]; ?></td>
                        <td><?php echo $row[7]; ?></td>
                        <td><?php echo $row[8]; ?></td>
                        <td><img src='../.../model/UserRegMgr/upload/<?php echo $row[12]; ?>' width="50" height="50"></td>

                        <td><br/>
                            <!-- direct to registration page and load correct user info based member id pass via url -->
                            <a href=UserRegister.php?user_memberId=<?php echo $row[1]; ?>><span style="color:blue" class=

                            <!-- delete the selected record -->
                            <a href='../.../model/UserRegMgr/_UserRegisterDelete.php?user_memberId=<?php echo $row[1]; ?>>

                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
    </div>
    <!-- END DEFAULT DATATABLE -->

```

```

        </div>
    </div>

    </div>
    <!--END PAGE CONTENT WRAPPER-->

    </div>
    <!--END PAGE CONTENT-->

    </div>
    <!-- END PAGE CONTAINER -->

    <!-- MESSAGE BOX-->
    <?php $template->getMessage(); ?>
    <!-- END MESSAGE BOX-->

    <!-- START SCRIPTS -->
    <?php $template->getScript(); ?>
    <!-- END SCRIPTS -->

</body>
</html>

```

Report Generation Coding

Coding segment of the report generation is as follows.

```

<?php
// get the HTML
ob_start();
?>
<link href="ReportStyle.css" rel="stylesheet" type="text/css"/>
<page style="font-size: 10pt">
    

    <br/><h1>Login Type Report</h1><br/><br/>
    <?php
    require_once '../control/config.php';

    $sql = 'SELECT login_type, count(login_type) login_typeC FROM db_slms.tbl_login group by login_type';

    $stmt = $conn->prepare($sql);
    $stmt->execute(array());

    echo '<table class="table" align="center">';
    echo '<thead>';
    echo '<tr>';
    echo '<th>login type</th>';
    echo '<th>number of members</th>';
    echo '</tr>';
    echo '</thead>';

    echo '<tbody>';
    while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
        echo '<tr>';
        echo '<td>' . $row[0] . '</td>';
        echo '<td>' . $row[1] . '</td>';
        echo '</tr>';
    }
    echo '</tbody>';
    echo '</table>';
    ?>
</page>

<?php
$content = ob_get_clean();

// convert in PDF
require_once('../lib/html2pdf/html2pdf.class.php');
try {
    $html2pdf = new HTML2PDF('P', 'A4', 'fr', true, 'UTF-8', 3);
    $html2pdf->pdf->SetDisplayMode('fullpage');
    $html2pdf->writeHTML($content, isset($_GET['vuehtml']));
    $html2pdf->Output('stuAttnRpt.pdf');
} catch (HTML2PDF_exception $e) {
    echo $e;
    exit;
}
?>

```


Creating a Line Chart Coding

To create a line chart, following code is used.

```
<?php
$sql = 'SELECT login_type, count(login_type) login_typeC FROM db_sims.tbl_login group by login_type';

$stmt = $conn->prepare($sql);
$stmt->execute(array());

while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
    $data1[] = array('login_type' => $row[0], 'login_typeC' => $row[1]);
}
?>

<script type="text/javascript">
var chart;
var chartData = <?php echo json_encode($data1); ?>;

AmCharts.ready(function () {

    // SERIAL CHART
    chart = new AmCharts.AmSerialChart();
    chart.pathToImages = "../amcharts/images/";
    chart.dataProvider = chartData;
    chart.categoryField = "login_type";
    chart.startDuration = 1;

    chart.handDrawn = true;
    chart.handDrawnScatter = 3;

    // AXES
    // category
    var categoryAxis = chart.categoryAxis;
    categoryAxis.gridPosition = "start";

    // value
    var valueAxis = new AmCharts.ValueAxis();
    valueAxis.axisAlpha = 0;
    chart.addValueAxis(valueAxis);

    // LINE
    var graph2 = new AmCharts.AmGraph();
    graph2.type = "line";
    graph2.title = "login type";
    graph2.lineColor = "#fcd202";
    graph2.valueField = "login_typeC";
    graph2.lineThickness = 3;
    graph2.bullet = "round";
    graph2.bulletBorderThickness = 3;
    graph2.bulletBorderColor = "#fcd202";
    graph2.bulletBorderAlpha = 1;
    graph2.bulletColor = "#ffffff";
    graph2.dashLengthField = "dashLengthLine";
    graph2.balloonText = "<span style='font-size:13px;'>[[title]] in [[category]]:<b>[[value]]</b> [[additional]]";
    chart.addGraph(graph2);

    // LEGEND
    var legend = new AmCharts.AmLegend();
    legend.useGraphSettings = true;
    chart.addLegend(legend);

    // WRITE
    chart.write("chartitemcopy");

});
</script>

</body>
</html>
```

Creating a Bar Chart Coding

To create a bar chart, following code is used.

```

</div>
<!-- END PAGE CONTAINER -->

<?php
$sql = 'SELECT login_type, count(login_type) login_typeC FROM db_slms.tbl_login group by login_type';

$stmt = $conn->prepare($sql);
$stmt->execute(array());

while ($row = $stmt->fetch(PDO::FETCH_NUM)) {

    $data[] = array('login_type' => $row[0], 'login_typeC' => $row[1]);
}
?>

<script type="text/javascript">
var chart;
var chartData = <?php echo json_encode($data); ?>;

AmCharts.ready(function () {
    // SERIAL CHART
    chart = new AmCharts.AmSerialChart();
    chart.pathToImages = "../amcharts/images/";
    chart.dataProvider = chartData;
    chart.categoryField = "login_type";
    chart.startDuration = 1;

    chart.handDrawn = true;
    chart.handDrawnScatter = 3;

    // AXES
    // category
    var categoryAxis = chart.categoryAxis;
    categoryAxis.gridPosition = "start";

    // value
    var valueAxis = new AmCharts.ValueAxis();
    valueAxis.axisAlpha = 0;
    chart.addValueAxis(valueAxis);

    // GRAPHS
    // column graph
    var graph1 = new AmCharts.AmGraph();
    graph1.type = "column";
    graph1.title = "login type";
    graph1.lineColor = "#a668d5";
    graph1.valueField = "login_typeC";
    graph1.lineAlpha = 1;
    graph1.fillAlpha = 1;
    graph1.dashLengthField = "dashLengthColumn";
    graph1.alphaField = "alpha";
    graph1.balloonText = "<span style='font-size:13px;'>[[title]] in [[category]]:<b>[[value]]</b> [[additional]]";
    chart.addGraph(graph1);

    // LEGEND
    var legend = new AmCharts.AmLegend();
    legend.useGraphSettings = true;
    chart.addLegend(legend);

    // WRITE
    chart.write("chartitemcopy");
});
</script>

</body>
</html>

```

Creating a Pie Chart Coding

By using pie charts, we can get the idea of an event at once. Coding which is used for creating pie chart is as follows.

```

<!-- PAGE TITLE -->
<div class="page-title">
    <h2><span class="fa fa-arrow-circle-o-left"></span> Login Type</h2>
</div>
<!-- END PAGE TITLE -->

<!--PAGE CONTENT WRAPPER-->
<div class="page-content-wrap">
    <div class="row">

        <div id="chartitemcopy" style="width:600px; height:400px;"></div>

    </div>

</div>
<!--END PAGE CONTENT WRAPPER-->

</div>
<!-- PAGE CONTENT -->

<!-- MESSAGE BOX-->
<?php $template->getMessage(); ?>
<!-- END MESSAGE BOX-->

<!-- START SCRIPTS -->
<?php $template->getScript(); ?>
<!-- END SCRIPTS -->

</div>
<!-- END PAGE CONTAINER -->

<?php
$sql = 'SELECT login_type, count(login_type) login_typeC FROM db_slms.tbl_login group by login_type';

$stmt = $conn->prepare($sql);
$stmt->execute(array());

while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
    $data1[] = array('login_type' => $row[0], 'login_typeC' => $row[1]);
}
?>

<script type="text/javascript">
    var chart;
    var chartData = <?php echo json_encode($data1); ?>;

    AmCharts.ready(function () {
        // PIE CHART
        chart = new AmCharts.AmPieChart();

        // title of the chart
        chart.addTitle("login type", 16);

        // title of the chart
        chart.addTitle("login type", 16);

        chart.dataProvider = chartData;
        chart.titleField = "login_type";
        chart.valueField = "login_typeC";
        chart.sequencedAnimation = true;
        chart.startEffect = "elastic";
        chart.innerRadius = "30%";
        chart.startDuration = 2;
        chart.labelRadius = 15;
        chart.balloonText = "[[title]]<br><span style='font-size:14px'><b>[[value]]</b> ([[percents]]%)</span>";
        // the following two lines makes the chart 3D
        chart.depth3D = 10;
        chart.angle = 15;

        // WRITE
        chart.write("chartitemcopy");
    });
</script>

</body>
</html>

```

Appendix G - Client Certificate



Vidyarthna University College Horana - Sri Lanka

Principal

Rev. Opalle Gnanasiri

B. A. (Hon's) , Dip. in Ed.

Tel: 034 - 2261268 (Office)

034 - 2261269 (Principal)

034 - 2261307 (Residence)

Date: 2017.10.19

Project Examination Board
University of Colombo School of Computing
No. 35, Reid Avenue
Colombo 07

Dear Sir/ Madam,

LETTER OF CERTIFICATION

This is to certify that Ms. B. R. Dilrukshi (BIT Registration number: R022201) has successfully developed Library Automation System for Vidyarthna University College, Horana for her BIT final year project. She gathered data and information from our school for development of the library system.

Vidyarthna University College has decided to accept and use her Library Automation System, since it is satisfying our requirements.

Thank you.

Your Sincerely,


.....
සමස්ත විද්‍යාලීන
වි. ප. පරීක්ෂණවලින්
B.com (SP) Dip in Ed.
(කම්පන විශ්වවිද්‍යාලය)
ස/කොරණ විද්‍යාලීන විශ්ව විද්‍යාලීන
කොරණ.

Figure G1: Client certificate

Glossary

Black-Box Testing

Test the external structure of the system.

Check-in

Return the library books.

Check-out

Take out books from the library.

Circulation

All the activities around the check-in and check-out process in the library.

CSS

Stand for “Cascading Style Sheet”. Use to adding styles to web documents.

Database

Pack of data or information that is organized for easy access.

Framework

It is a supporting structure for software development. It provides general functionality which can be change by additional user written code.

HTML

Stand for “Hyper Text Markup Language”. Use to build Web pages.

ILS

Stand for “Integrated Library System”.

Interface

Connect user to communicate with computer.

ISBN

Stand for “International Standard Book Numbering”. It is a unique book identifier.

JavaScript

Use to dynamic web applications. Develop by Netscape.

JQuery

It is a JavaScript library. It was designed to simplify the client-side scripting of HTML

MARC

Stand for “Machine Readable Cataloging”.

MySQL

One of the most popular database management systems which can handle large amount of data related to different data types.

PHP

Stand for “Hypertext Preprocessor”. Object oriented supported server side scripting language.

RUP

Stand for “Rational Unified process”.

SDLC

Stand for “System Development Life Cycle”.

SHA1

Stand for “Secure Hash Algorithm”.

SQL

Stand for “Structured Query Language”. Help to retrieve data base details.

UML

Stand for “Unified Modeling Language”.

Use case diagram

Use case diagram represent the how users interact with the system.

White box testing

Test the internal structure of the system.

XAMPP

Open source Bundled software package. Include Apache, PHP, MYSQL, and Perl.

Index

- Acceptance Testing, vi, 36
- Accuracy, 12
- accurate, 4
- action, 61
- Analysis, 3, 45
- architecture, 25
- Argo UML, 23, 24, 45
- Barcode, 10, 23, 47, 62
- Black-Box Testing, 35, 84
- Book Searching, v, 6
- Bootstrap, 23, 24, 46
- bottom, 48
- button, 48, 57, 61
- Calculate Fine, v, 6
- Card Catalogue, v, 6
- Cascading Style Sheets, iii
- case diagram, viii, 4, 5, 15, 45, 49, 85
- chapter, 4
- Check-in, viii, x, 38, 41, 42, 53, 54, 71, 84
- Check-out, viii, ix, x, 37, 40, 41, 51, 52, 68, 69, 70, 71, 84
- Circulation, v, ix, 6, 10, 13, 58, 60, 61, 62, 84
- Class Diagram, vi, 16
- Conclusion, 4
- Dashboard, 56
- database, 25, 48, 49, 77
- Database, 30, 48, 84
- databases, iii, 11, 26
- Design, vi, vii, 4, 18, 23, 45, 49
- developer, 44
- diagram, 85
- duplication, iii, 2
- Efficiency, 12
- Entity Relationship Diagram, vi, 16
- errors, 31
- Evaluation, vii, 4, 42, 70
- figure, 25, 56, 57, 58, 60, 61, 62, 70
- follows, 27, 73, 74, 75, 77, 79, 81
- Functional Requirements, v, 9, 11
- functions, 56
- Hardware, x, 23, 47
- Human Error, 2
- Hypertext markup language, iii
- Hypertext Pre-processor, iii, xi
- Implementation, vi, x, 4, 23
- important, 32
- information, 58
- Integration Testing, vi, 36
- International Standard Book Numbering, xi, 2, 84
- JavaScript, iii, 24, 25, 84
- knowledge, 2, 43, 44
- language, 44, 85
- logout, 57
- Machine Readable Catalogue, xi, 1
- Management, 67
- measure, 64
- member, 60
- Minimum, 47
- MVC, 25
- My SQL Workbench, 23, 24
- MySQL, iii, 24, 25, 26, 45, 47, 48, 84
- navigate, 48, 56, 58, 61, 62
- navigation, 58, 59, 64, 87
- Net Beans IDE, 23, 24, 46, 47
- Object Oriented Designing, vi, 14
- Performance*, 12
- performances, 3, 55, 64, 67
- PHP Data Objects, 26
- project, iv, 44
- provide, 3
- Rational Unified Process, iii, viii, 14
- Regression Testing, vii, 36
- Report Generation, v, 6, 79
- reports, 3, 44, 64
- Requirement gathering, 3
- requirements, 47
- school, 43, 44
- school library, 1, 3, 6, 7, 15, 16, 43
- Search, vi, viii, 10, 21, 37, 38, 40, 41, 51, 53, 58, 59, 61, 62, 63, 71
- Security, 1, 11
- Sequence Diagram, vi, 17
- server-side scripting, iii
- software, 2, 85
- Software, vi, x, 13, 23, 35, 45, 46, 47
- Software Testing, vi, 35
- structure, 85
- system, iv, 2, 3, 4, 25, 43, 44, 48, 56, 57, 60, 64, 72, 84, 85

System, 24, 47, 48
System Testing, vi, 36
teachers, 55, 56
Teachers, 56
Test Cases, vii, 36
understand, 49
Unit Testing, vi, 36

Usability, 11
User Authentication, 39, 70
User Interface, vi, 18
validation, 30, 31
verify, 48
White-Box Testing, 35
XAMPP, 24, 46, 47, 48, 85