



Faculty of Computer Science and Information Technology

***SMART LIBRARY MANAGEMENT SYSTEM  
(SLMS)***

**IQMAH SYAHIRAH BINTI SABRY**

Bachelor of Computer Science with Honours  
(Information System)

2025

**SMART LIBRARY MANAGEMENT SYSTEM  
(SLMS)**

**IQMAH SYAHIRAH BINTI SABRY**

This project is submitted in partial fulfilment of the  
requirements for the degree of  
Bachelor of Computer Science with Honours

Faculty of Computer Science and Information Technology  
UNIVERSITI MALAYSIA SARAWAK

2025

*Sistem Pengurusan Perpustakaan Pintar*  
(SLMS)

IQMAH SYAHIRAH BINTI SABRY

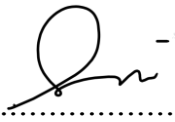
Projek ini merupakan salah satu keperluan untuk  
Ijazah Sarjana Muda Sains Komputer dengan Kepujian

Fakulti Sains Komputer dan Teknologi Maklumat  
UNIVERSITI MALAYSIA SARAWAK

2025

## DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

  
.....

Signature

Name : IQMAH SYAHIRAH BINTI SABRY

Matric No. : 78052

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 10/06/2025

## **ACKNOWLEDGMENT**

First and foremost, all praises and thanks be to Allah the Almighty, the Most Gracious and Most Merciful, for granting me the strength, wisdom, and guidance to complete this project successfully. Without His blessings, this achievement would not have been possible.

I would like to express my deepest gratitude to my supervisor, Dr. Fatihah binti Ramli for giving me the golden opportunity to undertake this project as my Final Year Project on the topic of the Smart Library Management System. I sincerely appreciate her patience, knowledge, and guidance throughout this project. Her support and encouragement have been invaluable in helping me successfully complete this work.

I am also grateful to Madam Hamizan binti Sharbini, my examiner, for her constructive feedback and valuable suggestions, which helped improve the quality of this project. Her insightful comments were instrumental in refining my work.

Additionally, I extend my appreciation to the Faculty of Computer Science and Information Technology for providing the necessary resources and a conducive environment for my research and project implementation.

My heartfelt thanks go to my parents for their endless love, prayers, and encouragement throughout this journey. Their unwavering belief in me has been my greatest motivation.

Finally, I would like to thank my friend Elyce for her assistance, moral support, and companionship, which helped me stay focused and complete this project on time.

## ABSTRACT

The Smart Library Management System (SLMS) is developed to address the lack of personalization, real-time notifications, and efficient management in traditional library systems. These systems often fail to provide tailored reading experiences and timely updates to users, leading to poor user engagement and operational inefficiencies for librarians.

This project presents a web-based solution that integrates genre-based filtering algorithms to offer personalized book recommendations based on users' borrowing and bookmarking behaviours. Real-time email notifications are implemented to keep users informed about due dates, request statuses, and new arrivals. The system also features an intuitive admin dashboard for managing book inventories and user interactions. Functional and usability testing demonstrated that users found the recommendations accurate and the interface user-friendly, with an 80% satisfaction rate.

The findings suggest that SLMS significantly enhances user experience and operational workflows in library management. Its implementation showcases how simple algorithmic personalization and timely notifications can transform traditional library environments into smart, engaging, and efficient digital systems.

**Keywords:** Smart Library Management System, genre-based filtering, book recommendation, real-time notification, user experience

## ABSTRAK

*Sistem Pengurusan Perpustakaan Pintar (SLMS) dibangunkan untuk menangani kekurangan personalisasi, pengesanan masa nyata, dan pengurusan yang efisien dalam sistem perpustakaan tradisional. Sistem-sistem ini sering gagal untuk menyediakan pengalaman pembacaan yang disesuaikan dan kemas kini yang tepat pada masanya kepada pengguna, yang menyebabkan penglibatan pengguna yang rendah dan ketidakcekapan operasi bagi pustakawan.*

*Projek ini membentangkan penyelesaian berasaskan web yang mengintegrasikan algoritma penapisan berdasarkan genre untuk menawarkan cadangan buku yang diperibadikan berdasarkan tingkah laku peminjaman dan penandaan pengguna. Pemberitahuan e-mel masa nyata dilaksanakan untuk memastikan pengguna dimaklumkan tentang tarikh akhir, status permintaan, dan ketibaan baru. Sistem ini juga mempunyai papan muka admin yang intuitif untuk mengurus inventori buku dan interaksi pengguna. Ujian fungsi dan kebolehgunaan menunjukkan bahawa pengguna mendapati cadangan itu tepat dan antara muka mesra pengguna, dengan kadar kepuasan 80%.*

*Penemuan menunjukkan bahawa SLMS secara signifikan meningkatkan pengalaman pengguna dan aliran kerja operasi dalam pengurusan perpustakaan. Pelaksanaannya menunjukkan bagaimana pemasangan algoritma peribadi yang sederhana dan notifikasi yang tepat pada masanya dapat mengubah persekitaran perpustakaan tradisional menjadi sistem digital yang pintar, menarik, dan efisien.*

**Kata kunci:** *Sistem Pengurusan Perpustakaan Pintar, penapisan berdasarkan genre, cadangan buku, notifikasi masa nyata, pengalaman pengguna*

## TABLE OF CONTENTS

|                                      |      |
|--------------------------------------|------|
| DECLARATION .....                    | iv   |
| ACKNOWLEDGMENT .....                 | v    |
| ABSTRACT .....                       | vi   |
| ABSTRAK.....                         | vii  |
| TABLE OF CONTENTS .....              | viii |
| LIST OF FIGURES .....                | xii  |
| LIST OF TABLES .....                 | xv   |
| LIST OF ABBREVIATIONS .....          | xvii |
| Chapter 1 .....                      | 1    |
| 1.1    Overview .....                | 1    |
| 1.2    Problem Statement.....        | 2    |
| 1.3    Project Aim .....             | 2    |
| 1.4    Objective.....                | 3    |
| 1.5    Methodology .....             | 4    |
| 1.6    Significance of Project ..... | 5    |
| 1.7    Project Schedule.....         | 6    |
| 1.8    Expected Outcome.....         | 6    |
| 1.9    Report Outline .....          | 7    |
| 1.10    Summary.....                 | 8    |
| Chapter 2 .....                      | 9    |
| 2.1    Overview .....                | 9    |

|                  |                                                       |           |
|------------------|-------------------------------------------------------|-----------|
| <b>2.2</b>       | <b>Review on similar projects .....</b>               | <b>9</b>  |
| 2.2.1            | Koha Integrated Library System (ILS).....             | 9         |
| 2.2.2            | Libsys Library Management System.....                 | 10        |
| 2.2.3            | Evergreen .....                                       | 11        |
| 2.2.4            | SLIM21 .....                                          | 11        |
| 2.2.5            | Smart Library System Universiti Teknologi MARA .....  | 12        |
| <b>2.3</b>       | <b>Key Technologies in Smart Library Systems.....</b> | <b>12</b> |
| 2.3.1            | Recommendation Engines .....                          | 12        |
| 2.3.2            | Real-time Notification Systems.....                   | 13        |
| 2.3.3            | Usage Trend Analysis.....                             | 13        |
| <b>2.4</b>       | <b>Gaps and Limitations in Existing Systems.....</b>  | <b>13</b> |
| <b>2.5</b>       | <b>Summary .....</b>                                  | <b>15</b> |
| <b>Chapter 3</b> | <b>.....</b>                                          | <b>17</b> |
| <b>3.1</b>       | <b>Overview .....</b>                                 | <b>17</b> |
| <b>3.2</b>       | <b>Methodology Phases .....</b>                       | <b>17</b> |
| 3.2.1            | Requirement Analysis.....                             | 17        |
| 3.2.1.1          | System Requirements .....                             | 18        |
| 3.2.1.2          | User Requirements.....                                | 18        |
| 3.2.2            | System Design .....                                   | 20        |
| 3.2.3            | Implementation.....                                   | 39        |
| 3.2.4            | Testing.....                                          | 40        |
| 3.2.5            | Maintenance.....                                      | 41        |

|                        |                                                                  |           |
|------------------------|------------------------------------------------------------------|-----------|
| 3.3                    | Summary .....                                                    | 41        |
| <b>CHAPTER 4 .....</b> |                                                                  | <b>42</b> |
| 4.1                    | Overview .....                                                   | 42        |
| 4.2                    | Installation and System Configuration.....                       | 42        |
| 4.2.1                  | Software Installation Procedure .....                            | 42        |
| 4.3                    | Functions of the System.....                                     | 45        |
| 4.3.1                  | Sign up Page .....                                               | 45        |
| 4.3.2                  | Sign-in Page .....                                               | 46        |
| 4.3.3                  | SLMS Homepage .....                                              | 48        |
| 4.3.4                  | Browse Book Page .....                                           | 49        |
| 4.3.5                  | View Book Page.....                                              | 50        |
| 4.3.6                  | Bookmark Page .....                                              | 51        |
| 4.3.7                  | Borrow Page and Read Page .....                                  | 52        |
| 4.3.8                  | Email Notifications for Borrowing Status and Due Date Alerts ... | 54        |
| 4.3.9                  | Notification Page .....                                          | 55        |
| 4.3.10                 | Admin Dashboard .....                                            | 56        |
| 4.3.11                 | Booklist Page .....                                              | 57        |
| 4.3.12                 | Request Page .....                                               | 57        |
| 4.3.13                 | Add Book Page.....                                               | 58        |
| 4.3.14                 | Manage Book Page.....                                            | 59        |
| 4.3.15                 | Genre-Based Filtering Algorithm Implementation and Results ...   | 60        |
| 4.4                    | Summary .....                                                    | 62        |

|                                                              |           |
|--------------------------------------------------------------|-----------|
| <b>CHAPTER 5 .....</b>                                       | <b>63</b> |
| <b>5.1 Overview .....</b>                                    | <b>63</b> |
| <b>5.2 Functional Testing .....</b>                          | <b>63</b> |
| <b>5.2.1 Unit Testing .....</b>                              | <b>63</b> |
| <b>5.3 Non-functional Testing.....</b>                       | <b>78</b> |
| <b>5.3.1 Usability Testing.....</b>                          | <b>78</b> |
| <b>5.4 Summary .....</b>                                     | <b>81</b> |
| <b>Chapter 6 .....</b>                                       | <b>82</b> |
| <b>6.1 Overview .....</b>                                    | <b>82</b> |
| <b>6.2 Project Achievements .....</b>                        | <b>82</b> |
| <b>6.3 Project Limitations .....</b>                         | <b>83</b> |
| <b>6.4 Future Works.....</b>                                 | <b>83</b> |
| <b>6.5 Summary .....</b>                                     | <b>84</b> |
| <b>Reference.....</b>                                        | <b>85</b> |
| <b>Appendix A: Project Schedule and Gantt Chart.....</b>     | <b>87</b> |
| <b>Appendix B: Questionnaire for Usability Testing .....</b> | <b>89</b> |

## LIST OF FIGURES

|                                                             |    |
|-------------------------------------------------------------|----|
| Figure 1.1: Incremental Method .....                        | 4  |
| Figure 2.1: UI for Koha Library Software .....              | 10 |
| Figure 2.2: UI for Libsys Library Management System.....    | 10 |
| Figure 2.3: UI for Evergreen ILS .....                      | 11 |
| Figure 2.4: UI for iSLIM - Library Management System.....   | 12 |
| Figure 3.1: UCD.....                                        | 22 |
| Figure 3.2: Activity Diagram for User .....                 | 27 |
| Figure 3.3: Activity Diagram for Admin .....                | 28 |
| Figure 3.4: Sequence Diagram for User Login.....            | 29 |
| Figure 3.5: Sequence Diagram for Book Borrowing .....       | 29 |
| Figure 3.6: Sequence Diagram for Admin Adding a Book.....   | 30 |
| Figure 3.7: Sequence Diagram for Book Recommendation .....  | 30 |
| Figure 3.8: Sequence Diagram for Due Date Notification..... | 31 |
| Figure 3.9: Class Diagram.....                              | 32 |
| Figure 3.10: Sign-in Page .....                             | 33 |
| Figure 3.11: Sign Up Page.....                              | 34 |
| Figure 3.12: Homepage.....                                  | 34 |
| Figure 3.13: Recommended Book for User.....                 | 35 |
| Figure 3.0.14: Filter Books.....                            | 35 |
| Figure 3.15: View Book Page.....                            | 36 |
| Figure 3.16: Notification Page.....                         | 36 |
| Figure 3.17: Dashboard .....                                | 37 |
| Figure 3.18: Manage Book.....                               | 37 |
| Figure 3.19: Booklist .....                                 | 38 |

|                                                   |    |
|---------------------------------------------------|----|
| Figure 3.20: Manage Requests .....                | 38 |
| Figure 4.1: Librarywebsite Table.....             | 44 |
| Figure 4.2: Smart Library Management System.....  | 44 |
| Figure 4.3: Sign up Page .....                    | 45 |
| Figure 4.4: Username or Email Already Exists..... | 46 |
| Figure 4.5: Sign in Page .....                    | 47 |
| Figure 4.6: Error Username Not Found.....         | 47 |
| Figure 4.7: Error Incorrect Password .....        | 47 |
| Figure 4.8: Homepage.....                         | 48 |
| Figure 4.9: Trending Books.....                   | 48 |
| Figure 4.10: Inspirational Quotes .....           | 49 |
| Figure 4.11: Recommended Books and Footer.....    | 49 |
| Figure 4.12: Browse Page.....                     | 50 |
| Figure 4.13: View Book Page.....                  | 51 |
| Figure 4.14: Related Books .....                  | 51 |
| Figure 4.15: Bookmark Page.....                   | 52 |
| Figure 4.16: Borrow Page.....                     | 53 |
| Figure 4.17: Request submitted .....              | 53 |
| Figure 4.18: Read Page.....                       | 53 |
| Figure 4.19: Book Request Accepted.....           | 54 |
| Figure 4.20: Reminder.....                        | 55 |
| Figure 4.21: New Book Notification.....           | 55 |
| Figure 4.22: Notification Page.....               | 56 |
| Figure 4.23: Admin Dashboard .....                | 56 |
| Figure 4.24: Booklist Page .....                  | 57 |
| Figure 4.25: Requests Page.....                   | 58 |

|                                                 |    |
|-------------------------------------------------|----|
| Figure 4.26: Notification for Book Request..... | 58 |
| Figure 4.27: Add Book Page .....                | 59 |
| Figure 4.28: Manage Book Page.....              | 60 |
| Figure 5.1: Suggestions for Improvement.....    | 80 |

## LIST OF TABLES

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| Table 2.1: Gaps and Limitations in Existing Systems .....                        | 15 |
| Table 3.1: Software Requirements .....                                           | 18 |
| Table 3.2: Functional Requirements .....                                         | 19 |
| Table 3.3: System Automated Features .....                                       | 19 |
| Table 3.4: Non-Functional Requirements .....                                     | 19 |
| Table 3.5: User Roles .....                                                      | 20 |
| Table 3.6: Use case explanation for the "Sign Up" feature .....                  | 23 |
| Table 3.7: Use case explanation for the "Sign In" feature .....                  | 23 |
| Table 3.8: Use case explanation for the "Browse book" feature .....              | 23 |
| Table 3.9: Use case explanation for the "Borrow/Recommended Books" feature ..... | 24 |
| Table 3.10: Use case explanation for the "View Recommended Books" feature .....  | 24 |
| Table 3.11: Use case explanation for the "Return Physical Book" feature .....    | 25 |
| Table 3.12: Use case explanation for the "View Dashboard" feature .....          | 25 |
| Table 3.13: Use case explanation for the "Add/Edit Book" feature .....           | 25 |
| Table 3.14: Use case explanation for the "Manage Requests" feature .....         | 26 |
| Table 3.15: Use case explanation for the "Notify Genre" feature .....            | 26 |
| Table 5.1: Test Case for Register .....                                          | 64 |
| Table 5.2: Test Case for Sign-in .....                                           | 65 |
| Table 5.3: Test Case for Homepage .....                                          | 66 |
| Table 5.4: Test Case for Browse Book .....                                       | 67 |
| Table 5.5: Test Case for View Book .....                                         | 68 |
| Table 5.6: Test Case for Bookmark .....                                          | 69 |
| Table 5.7: Test Case for Borrow and Read .....                                   | 70 |
| Table 5.8: Test Case for Recommendation Functionality .....                      | 71 |

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| Table 5.9: Email Notifications for Borrowing Status and Due Date Alerts ..... | 72 |
| Table 5.10: Test Case for Notification Page.....                              | 73 |
| Table 5.11: Admin Dashboard.....                                              | 74 |
| Table 5.12: Test Case for Booklist.....                                       | 74 |
| Table 5.13: Test Case for Manage Requests .....                               | 75 |
| Table 5.14: Test Case for Add Book .....                                      | 76 |
| Table 5.15: Test Case for Manage Book.....                                    | 76 |
| Table 5.16: Functionality of the System .....                                 | 79 |
| Table 5.17: User Interface and Design.....                                    | 79 |
| Table 5.18: Overall System .....                                              | 80 |
| Table 6.1:Project objectives and project achievements.....                    | 82 |

## LIST OF ABBREVIATIONS

| Abbreviations | Descriptions                    |
|---------------|---------------------------------|
| SDLC          | Software Development Life Cycle |
| SLMS          | Smart Library Management system |
| UML           | Unified Modeling Language       |
| UCD           | Use Case Diagram                |
| UI            | User Interface                  |
| UX            | User Experience                 |
| VS Code       | Visual Studio Code              |

## **Chapter 1**

### **INTRODUCTION**

#### **1.1 Overview**

The Smart Library Management System (SLMS) is a web-based application designed to enhance the library experience by integrating personalized book recommendations, real-time notifications, and efficient library management functionalities. The system is carefully designed to support seamless data flow and structured database management, ensuring a smooth user experience while maintaining accurate and organized records.

One of the core features of SLMS is the personalized recommendation system, which suggests books based on user preferences, borrowing history, and bookmarking behavior. This enhances user engagement by offering tailored reading suggestions, that make book discovery more intuitive and relevant. The system also incorporates real-time updates and notifications to keep users informed about new books matching their interests, current book availability, and reminders for returning borrowed books on time.

For administrators, the system offers essential functionalities such as adding, editing, and deleting book records. Additionally, the system features a user-friendly administrative dashboard designed specifically for librarians. This dashboard offers a comprehensive overview of library activities, enabling efficient monitoring of library statistics, streamlined management of book inventory, and effective processing of user borrowing requests. These functionalities support better resource management and informed decision-making. Through these features, the SLMS aims to create a dynamic, engaging, and responsive library environment that meets the evolving needs of its users.

## **1.2 Problem Statement**

Traditional library management systems often fall short in offering personalization, real-time updates, and efficient data handling, which are essential for enhancing user experience and streamlining library operations. Without support for data-driven insights, libraries struggle to provide personalized book recommendations, resulting in lower user engagement and ineffective book discovery.

Also, without real-time notifications about new books, book availability, and due dates, users may miss important updates, making it harder and slower for them to find and borrow books. For librarians, not having key features like managing borrowed books, keeping track of the book inventory, sending return reminders, and updating users about new arrivals makes it harder to run the library smoothly and efficiently.

These challenges highlight the need for a comprehensive SLMS that integrates personalized recommendations, real-time notifications, and efficient administrative tools to foster a more engaging, accessible, and well-managed library environment.

## **1.3 Project Aim**

The SLMS aims to enhance the overall library experience by integrating personalized book recommendations, real-time notifications, and efficient library management within a well-structured and reliable system architecture.

The system will be designed with seamless data flow and an organized database to support personalized recommendations based on user interests, borrowing history, and bookmarking behavior. To boost user engagement, SLMS will provide real-time notifications that keep users informed about book availability, new arrivals matching their preferences, and timely reminders for returning borrowed books.

Furthermore, the system will include comprehensive administrative modules that enable librarians to efficiently manage borrowed books, monitor inventory levels, send return reminders, and make data-driven decisions related to book acquisitions and resource allocation.

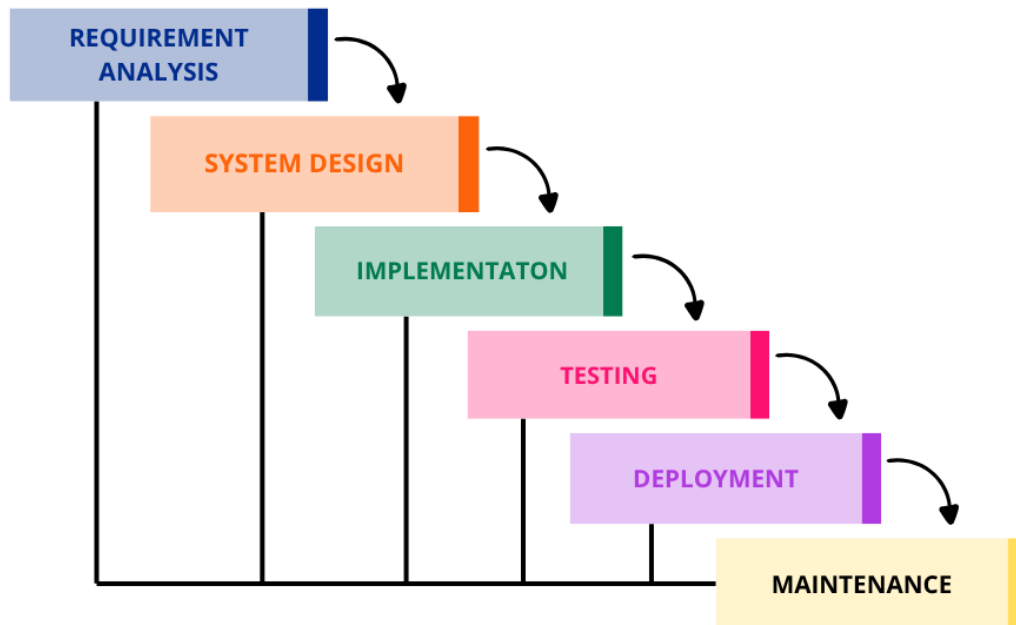
By integrating these features, SLMS aims to modernize library operations, improve user satisfaction, and create a dynamic, accessible, and responsive library environment.

#### **1.4 Objective**

- i. To design the system and database so it can provide personalized book recommendations based on what users like, borrow, and bookmark.
- ii. To develop key features for personalized recommendations, real-time notifications, and easy management of borrowed books and book records.
- iii. To provide librarians with a user-friendly system to monitor library statistics, manage book inventory, and process user borrowing requests through dedicated administrative modules.

## 1.5 Methodology

# INCREMENTAL MODEL



*Figure 1.1: Incremental Method*

The development of SLMS will follow the Incremental Model methodology as shown in Figure 1.1, where the system is built and delivered in smaller parts through multiple development cycles. This approach allows for manageable development stages, focusing on core features first and gradually adding additional functionalities (Sommerville, 2016).

In the initial phase, system requirements will be gathered and analysed based on existing library needs. The system architecture will be designed to support personalized book recommendations through well-structured data flow and database design.

The development phase will be divided into increments, starting with the implementation of essential features such as user management, book recommendations, and borrowing management. Subsequent increments will add real-time notifications and

administrative dashboards. Each increment will undergo testing to ensure functionality and reliability before moving to the next phase (Sommerville, 2016).

Prototyping tools like Figma will be used to design user interfaces, and development will be carried out using Visual Studio. Testing and debugging will be performed continuously to refine the system and improve user experience.

The final phase will focus on full system integration, rigorous testing, documentation, and deployment. This incremental approach ensures a structured yet flexible development process that results in a robust and user-friendly SLMS.

## **1.6 Significance of Project**

SLMS aims to modernize library operations by integrating genre-based content-based filtering algorithms to provide personalized book recommendations, real-time updates, and efficient management of library resources. By replacing manual processes with this technique, the SLMS can suggest books tailored to users' preferences, borrowing history, and bookmarking behavior, as well as current popular trends.

For users, SLMS offers more engaging and customized book discovery experience through personalized recommendations and real-time notifications. These notifications keep users informed about new books matching their interests, current book availability, and timely reminders for returning borrowed books, helping users save time and stay connected with the library's offerings.

For librarians, the system includes comprehensive administrative modules that enable efficient management of borrowed books, monitoring of inventory levels, sending return reminders, and making data-driven decisions related to book acquisitions and resource allocation. The genre-based filtering approach simplifies these tasks, providing

practical and accessible tools for library staff without the need for complex machine-learning models.

By bridging the gap between traditional library systems and modern genre-based content-based filtering automation, the SLMS supports a more efficient, user-friendly, and responsive library environment that benefits both users and administrators.

### **1.7 Project Schedule**

This project is planned to be completed over two semesters, covering the Final Year Project I and Final Year Project II sessions for the academic year 2024/2025. Final Year Project I (FYP1) will include chapters 1 Introduction, 2 Literature Review, and 3 Design and Methodology. Meanwhile, Final Year Project II (FYP2) will comprise chapters 4 Implementation, 5 Testing, and 6 Conclusion. The detailed project timeline can be found in Appendix A.

### **1.8 Expected Outcome**

The SLMS is expected to deliver a well-organized and reliable system that supports personalized book recommendations and efficient library management. Using genre-based filtering algorithms, the system will provide tailored book suggestions based on users' borrowing history, and bookmarking behavior, making book discovery more relevant and engaging.

The system will also feature real-time notifications to keep users informed about new books matching their preferences, current book availability, and timely reminders for returning borrowed books. Enhanced search functions will help users easily find books through category filters and keyword searches.

For librarians, SLMS will include comprehensive administrative modules to manage borrowed books, monitor inventory levels, send return reminders, and make data-driven decisions regarding book acquisitions and resource allocation. By integrating these

features, the system aims to improve user satisfaction and streamline library operations, creating a modern, accessible, and responsive library environment.

## **1.9 Report Outline**

Chapter 1 introduces the topic and provides a detailed discussion covering the introduction, problem statement, project aim, objectives, methodology, significance of the project, project schedule, and expected outcomes.

Chapter 2 reviews relevant literature collected from various journals, papers, and articles related to the project. It summarizes existing studies and research to compare and evaluate the features and limitations of similar projects.

Chapter 3 focuses on the chosen methodology, which is the Incremental Model, used as a framework for system development. This chapter details the activities carried out in each phase and includes key UML diagrams such as the UCD, Sequence Diagram, and Class Diagram to illustrate the system's structure and interactions. It also presents the prototype interfaces of the project.

Chapter 4 discusses the implementation of the proposed system. It describes the interface design to provide a clear understanding of the system's layout and flow. This chapter also covers the tools used during the development process.

Chapter 5 is dedicated to testing the proposed system. Both functional and non-functional requirements will be verified to ensure the system meets specifications. User testing will be conducted, and the results will be analysed and discussed.

Finally, Chapter 6 concludes the report by summarizing the entire project. It also addresses the limitations of the proposed system and suggests areas for future improvement.

## **1.10 Summary**

This chapter explains the key concepts utilized in the SLMS, with a focus on the chosen algorithm, genre-based filtering, for the system's implementation. It outlines the project's objectives and highlights its significance. Additionally, a brief overview of the methodology to be applied in developing the web application is provided.

## **Chapter 2**

### **LITERATURE REVIEW**

#### **2.1 Overview**

This chapter presents a review of relevant literature, focusing on the analysis, comparison, and evaluation of selected studies and projects related to SLMS. It explores the techniques and methodologies implemented in these systems, particularly those incorporating intelligent features such as book recommendation engines, real-time notifications, and usage trend analysis. The chapter concludes with a summary of key insights gained from the reviewed works.

#### **2.2 Review on similar projects**

Several library systems and digital platforms have been developed over the years to improve the management, accessibility, and personalization of library services. This section reviews notable systems and projects that share similarities with the proposed Smart Library Management System (SLMS), analyzing their strengths, limitations, and relevance to this project.

##### **2.2.1 Koha Integrated Library System (ILS)**

Koha is one of the most widely adopted open-source integrated library systems, used by academic and public libraries worldwide. It offers features such as cataloguing, circulation, acquisitions, and patron management through a web-based interface. While Koha is reliable and scalable, it lacks modern smart features such as personalized book recommendations and real-time notifications (Ashikuzzaman, 2024). Furthermore, its user interface, though functional, is not highly customizable or intuitive for modern user expectations.

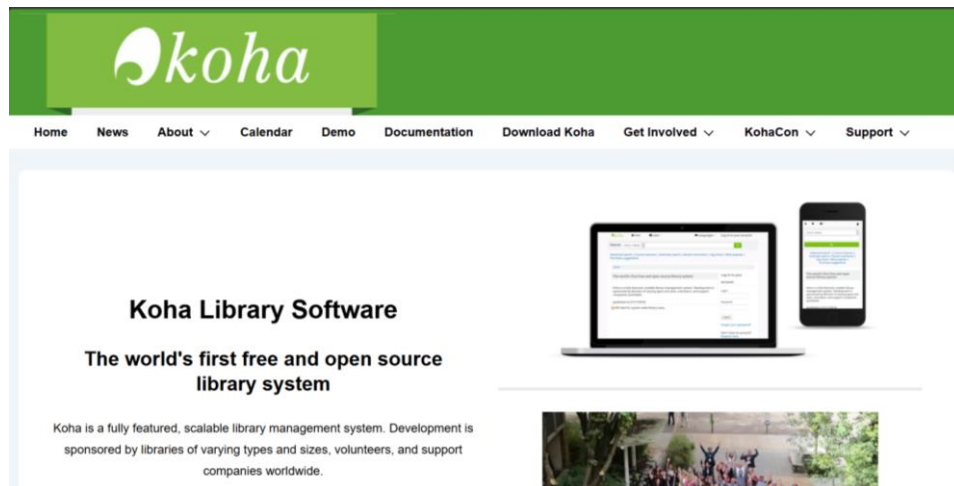


Figure 2.1: UI for Koha Library Software

## 2.2.2 Libsys Library Management System

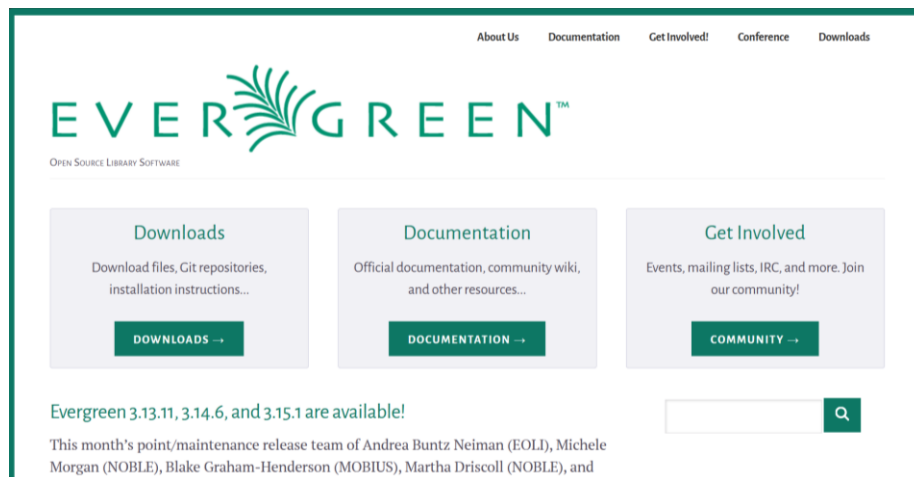
Libsys is a proprietary system widely used in Indian academic institutions. It supports advanced cataloguing, barcode generation, circulation tracking, and inventory control. While Libsys is known for its comprehensive features and reliability, as described by Krishnappa and G (2023), its closed-source nature presents a limitation by restricting customization and integration with third-party tools such as recommendation engines and notification services. It is better suited for institutions with larger budgets and technical support teams.



Figure 2.2: UI for Libsys Library Management System

### 2.2.3 Evergreen

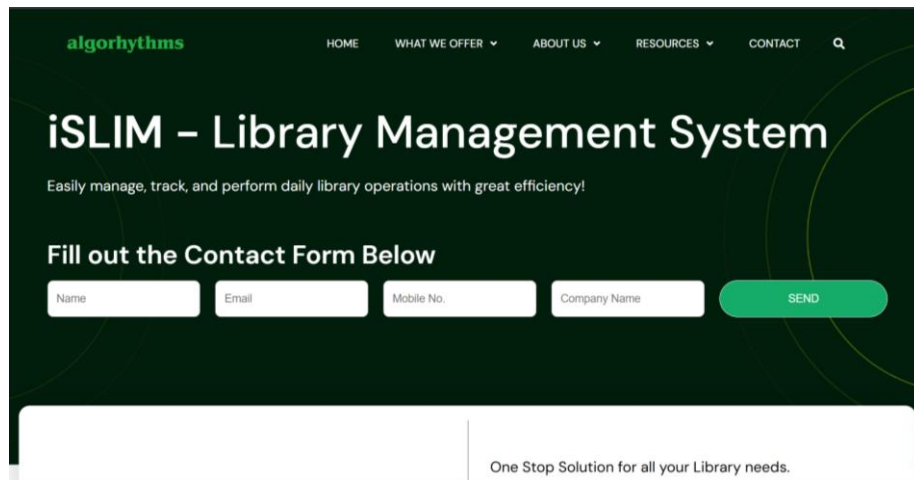
Evergreen is an open-source Integrated Library System (ILS) developed for libraries in Georgia, USA. It offers features like user account management, barcode tracking, fine calculation, and search optimization. It is scalable for library consortia but does not natively include AI-powered recommendations or user behavior analytics (Wikipedia, 2023). Notifications are mostly limited to due-date reminders via email.



*Figure 2.3: UI for Evergreen ILS*

### 2.2.4 SLIM21

System for Library Information Manage (SLIM21) now known as iSLIM is a commercial library management software used in Southeast Asia. It includes features like digital asset management and automated circulation. Although it has a modern interface and cloud-based access, it does not integrate intelligent recommendation systems or usage trend analysis tools (Janaji Jadhav & Aanada Kamble, 2022). Its primary focus is on managing digital and physical collections rather than enhancing user engagement.



*Figure 2.4: UI for iSLIM - Library Management System*

### **2.2.5 Smart Library System Universiti Teknologi MARA**

A research project by UiTM developed a Smart Library System prototype with RFID tracking, self-service kiosks, and user notifications via SMS/email (Khusaini et al., 2023). However, the system lacked advanced recommendation features and was more focused on automation of circulation processes.

## **2.3 Key Technologies in Smart Library Systems**

Smart Library Management Systems integrate several advanced technologies to improve user experience and operational efficiency. Three critical technologies are recommendation engines, real-time notification systems, and usage trend analysis.

### **2.3.1 Recommendation Engines**

Personalized book recommendations are increasingly important in modern library systems. These engines typically use content-based filtering, collaborative filtering, or hybrid approaches to suggest books relevant to users' preferences and past interactions. Content-based filtering analyzes metadata such as genre, author, and keywords to recommend similar items, which is particularly effective in cold-start situations where user data is limited (The Upwork Team, 2021). Unlike traditional library systems

reviewed, many lack sophisticated recommendation features that enhance book discovery and user satisfaction.

### **2.3.2 Real-time Notification Systems**

Timely communication about book availability, due dates, and item status updates plays an important role in enhancing user engagement within library systems. Real-time notifications can be delivered through various channels such as email, SMS, mobile push notifications, or in-app alerts. While some systems, like the UiTM Smart Library prototype, have adopted SMS and email alerts to automate circulation updates (Khusaini et al., 2023), many existing integrated library systems such as Koha and Evergreen primarily offer basic email notifications. These notifications are typically limited to due date reminders, overdue notices, and hold availability alerts, with no built-in support for personalized or interest-based updates on new arrivals (Guarracino, 2023).

### **2.3.3 Usage Trend Analysis**

**Data-driven library management** can enhance decision-making by using analytics to track borrowing trends and user preferences. However, many existing systems focus mainly on basic inventory control and circulation tracking, without offering simple, centralized tools for day-to-day administrative tasks. This limits the librarian's ability to efficiently manage operations. The proposed system addresses this by providing a user-friendly administrative module to manage book records, process user borrowing requests, and monitor basic system activity supporting smoother library operations, even without complex analytics.

## **2.4 Gaps and Limitations in Existing Systems**

Despite the advances made by existing library management systems, several gaps and limitations persist that hinder the full realization of a truly smart and user-centric library experience.

Firstly, many widely-used systems such as Koha and Evergreen, though reliable and scalable, lack modern intelligent features like personalized book recommendation engines. These features are critical in enhancing book discovery and tailoring suggestions to individual user preferences, which improves user satisfaction and engagement. Without smart recommendation systems, users usually must search for books themselves or get general suggestions, which makes the experience less personal and helpful.

Secondly, the real-time notification features in many library systems remain basic and limited. Most platforms only offer standard email reminders for due dates, overdue items, or reservation availability. This limitation reduces the immediacy and effectiveness of user communication, which may lead to missed updates, delayed actions, and lower overall user engagement.

Thirdly, many existing library systems lack a centralized and user-friendly interface for administrative tasks. While they may support circulation tracking, they often do not provide a simple way for librarians to manage book inventory, handle borrowing requests, or monitor basic system activity in one place. This can slow down workflows and reduce efficiency in managing daily operations.

Lastly, many library systems like Libsys are closed-source, meaning users cannot change or customize the system freely. This makes it hard to add new features or connect the system with other tools. For smaller libraries with limited technical support, this can be a challenge when they want to adjust the system to fit their specific needs.

The following Table 1.1 summarizes these gaps and limitations for the reviewed systems:

Table 2.1: Gaps and Limitations in Existing Systems

| Library System            | Personalized Recommendation | Real-time Notifications                                     | Usage Trend Analysis  | Customization & Integration                               |
|---------------------------|-----------------------------|-------------------------------------------------------------|-----------------------|-----------------------------------------------------------|
| Koha                      | No                          | Basic email reminders about due dates and overdue notices   | Limited               | Open-source but UI not highly customizable                |
| Libsys                    | No                          | Basic email reminders about due dates and overdue notices   | Limited               | Proprietary, limited customization                        |
| Evergreen                 | No                          | Basic email reminders about due dates and overdue notices   | Limited               | Open-source, integration possible but limited AI features |
| SLIM21 (iSLIM)            | No                          | Basic email reminders about due dates and overdue notices   | No advanced analytics | Commercial, limited integration                           |
| UiTM Smart Library System | Limited                     | SMS/Email notifications about due dates and overdue notices | No advanced analytics | Research prototype, limited scope                         |

## 2.5 Summary

In summary, this chapter presented a comprehensive review of existing library management systems and the smart technologies integrated into them. While systems such as Koha, Libsys, Evergreen, SLIM21, and UiTM's Smart Library prototype offer valuable functionalities for cataloguing, circulation, and inventory control, they exhibit notable limitations in terms of intelligent features. Specifically, the absence of personalized recommendation engines, limited real-time notification capabilities, and a lack of in-depth usage trend analytics highlight a significant gap in current solutions.

Furthermore, the analysis of key smart technologies such as recommendation systems, real-time notifications, and usage analytics demonstrates their potential in transforming traditional libraries into intelligent, user-centered environments. However,

current implementations either lack these features or only adopt them partially without deep integration.

This literature review shows why the proposed SLMS is needed. By improving the weaknesses found in current systems, the SLMS aims to offer a smarter, more user-friendly, and efficient library experience for both users and librarians.

## **Chapter 3**

### **DESIGN AND METHODOLOGY**

#### **3.1 Overview**

This chapter outlines the methodology selected for the development of this project, which is the Incremental Model within the SDLC framework. The project's proposed requirements will be detailed during the requirement analysis and design phase. Each phase of the development process will be discussed comprehensively throughout this chapter.

#### **3.2 Methodology Phases**

The Incremental Model is chosen as the methodology to develop the SLMS. This approach involves breaking down the system into smaller, manageable modules that are developed and delivered in increments (Ali, 2020). Each increment adds new functionality to the system based on user requirements and feedback. Unlike the linear flow of the Waterfall Model, the Incremental Model allows for gradual development and continuous improvement. The main phases involved include requirement analysis, system design, incremental implementation, testing, integration, and evaluation. This iterative process ensures that key features such as book recommendations, real-time notifications, and usage analytics can be developed, tested, and refined progressively. The incremental development approach enables these key features such as personalized recommendations, real-time notifications, and admin management tools to be built and refined in stages, ensuring continuous improvement and alignment with user needs throughout the development process.

##### **3.2.1 Requirement Analysis**

This phase involves identifying and understanding the requirements necessary for the development of the SLMS. Since this project primarily focuses on prototyping using

Figma, the requirements were established through research and comparative analysis of existing digital library systems, as well as by applying best practices in user interface and user experience design. No user surveys or direct feedback were conducted during this phase. Instead, system functionalities were defined based on anticipated user needs and the overall objective of improving the digital library experience. The requirements identified for the SLMS are categorized into two groups: system requirements and user requirements, which include functional requirements, automated functional features, non-functional requirements, and user roles.

### 3.2.1.1 System Requirements

This section outlines the essential software and hardware requirements needed for the development of the proposed system. Table 3.1 presents the software requirements for this project in detail.

*Table 3.1: Software Requirements*

| Software           | Description                                                                |
|--------------------|----------------------------------------------------------------------------|
| Draw.io            | Tool for creating UML diagrams and flowcharts.                             |
| Figma              | Used to design the prototype interface (UI/UX) of the SLMS.                |
| MySQL              | Relational database to store system data such as books, users, and genres. |
| PhpMyAdmin         | Web-based interface to manage MySQL database easily.                       |
| Visual Studio Code | Code editor used to develop the SLMS using PHP, HTML, CSS, and JavaScript. |
| Web Browser        | To view and test the web-based system during development.                  |
| XAMPP              | Local server environment to run PHP and MySQL.                             |

### 3.2.1.2 User Requirements

This section outlines the user requirements necessary for the development of the proposed system. Table 3.2 presents the functional requirements of the system, Table 3.3 highlights the automated functional features, which operate independently to enhance

user experience and system efficiency, Table 3.3 details the non-functional requirements, and Table 3.4 outlines the user roles and their interactions with the system.

*Table 3.2: Functional Requirements*

| <b>Functional Requirement</b>        | <b>Description</b>                                                                                       |
|--------------------------------------|----------------------------------------------------------------------------------------------------------|
| User Registration and Authentication | Allows users to register, log in, and log out securely.                                                  |
| Book Browsing and Search             | Enables users to browse the library catalogue and search for books by title, author, genre, or keywords. |
| Personalized Book Recommendations    | Provides tailored book suggestions based on users' borrowing history and preferences.                    |
| Real-Time Notifications              | Sends timely alerts to users about book availability, due dates, and new arrivals.                       |
| Admin Management                     | Enables administrators to add, edit, or delete book records and manage user accounts.                    |
| Dashboard Overview                   | Gives administrators a visual summary of system activity and key statistics.                             |
| Bookmarking and History Tracking     | Allows users to bookmark favourite books and view their borrowing history.                               |

*Table 3.3: System Automated Features*

| <b>Feature</b>           | <b>Description</b>                                                                                         |
|--------------------------|------------------------------------------------------------------------------------------------------------|
| New Book Notifications   | Automatically notifies users when a new book matches their preferred genre.                                |
| Book Availability Alerts | Sends notifications when a previously unavailable book becomes available, after user requests "Notify Me." |
| Real-Time Status Updates | Instantly updates book availability after a borrowing request is approved or a book is returned.           |

*Table 3.4: Non-Functional Requirements*

| <b>Non-Functional Requirement</b> | <b>Description</b>                                                                              |
|-----------------------------------|-------------------------------------------------------------------------------------------------|
| Interface                         | The design should be simple and easy to use, with clear wording so users can interact smoothly. |
| Usability                         | The system should be user-friendly and accessible to users of all skill levels.                 |

|               |                                                                                              |
|---------------|----------------------------------------------------------------------------------------------|
| Validation    | The system should verify user input and provide immediate feedback if errors are detected.   |
| Accessibility | The system should offer different views or options based on user roles to enhance usability. |
| Availability  | The system should always remain accessible to users with minimal downtime.                   |

*Table 3.5: User Roles*

| User Roles    | Description                                                                                                                                                                                                                                                                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| User          | Users can register an account, browse and search for books, receive personalized book recommendations, bookmark their favourite books, and get notifications such as reminders for returning borrowed books.                                                                                                                                                           |
| Administrator | Administrators have the authority to manage book records and oversee borrowing requests. Their responsibilities include adding new books, handling user borrow requests by approving or rejecting them, reclaiming books, sending return reminders, editing book details, viewing the list of books, and accessing an overall dashboard to monitor library activities. |

### 3.2.2 System Design

The system design will be completed before any coding is carried out, as it defines how the system will function and ensures that it aligns with user requirements. In this section, the system architecture will be explained in detail using several UML diagrams, including the UCD, Sequence Diagram, Activity Diagram, and Class Diagram, which are developed using Draw.io. These diagrams help illustrate the system's functionality, flow, and structure. Additionally, a high-fidelity prototype of the system's user interface will be designed using Figma to visualize the layout, interactions, and content of the SLMS.

#### 3.2.2.1 Use Case Diagram (UCD)

A UCD is responsible for visualizing the interactions between external entities such as users or other systems and the functionalities of the system (Parker, 2019). It illustrates the main functionalities of the system and how users interact with them. In

the proposed system, users must first register and log in before they can access features such as browsing, searching for books, and entering their preferred book information. Based on their input, the system will generate personalized book recommendations. Figure 3.1 presents the use case diagram for the proposed system.

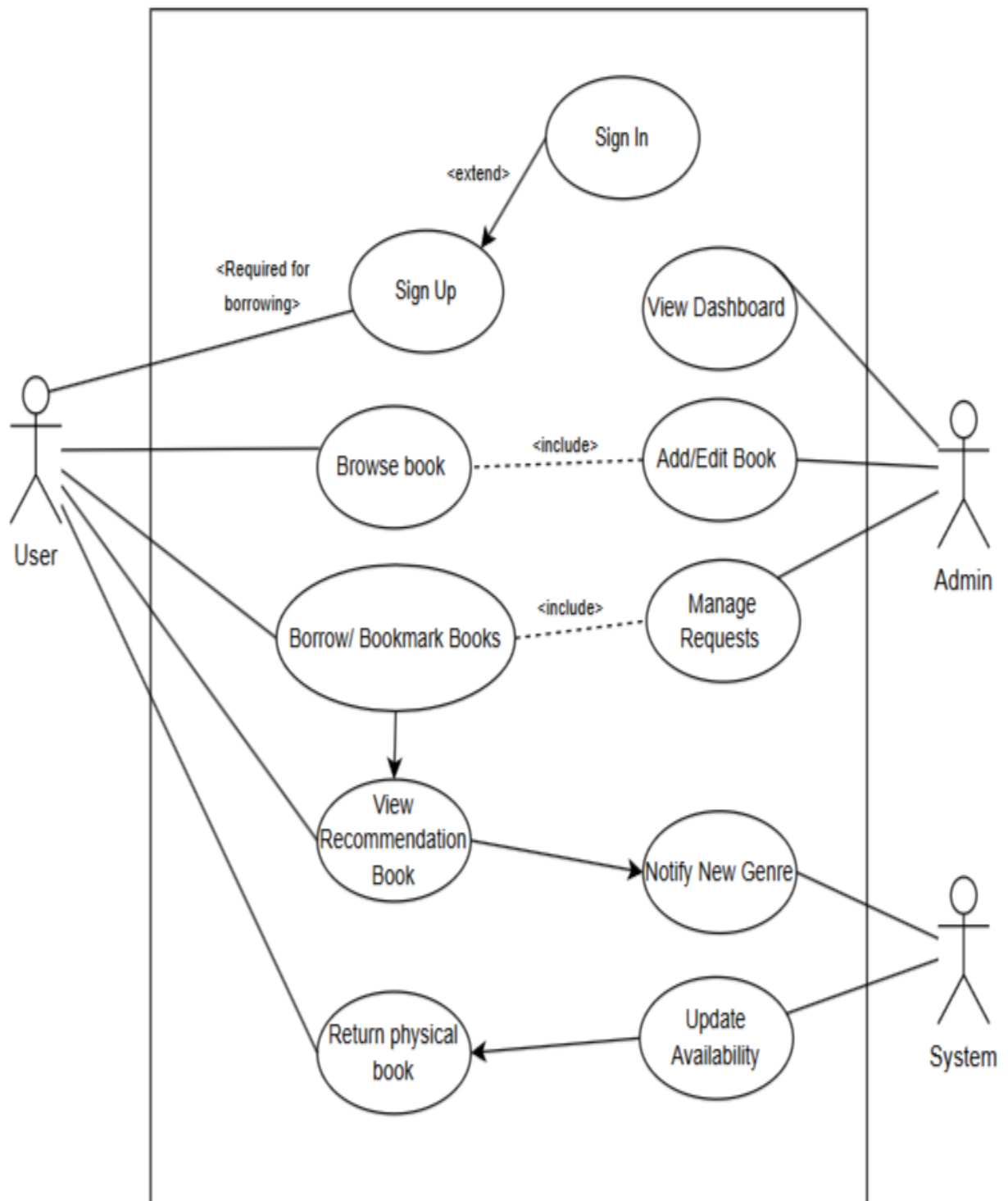


Figure 3.1: UCD

Referring to the use case in Figure 3.1, the description of each use cases is described as follows;

*Table 3.6: Use case explanation for the "Sign Up" feature*

|                     |                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name       | Sign Up                                                                                                                                                                                                                                                        |
| Description         | Allows new users to register by providing personal details like name, email, and password.                                                                                                                                                                     |
| Actor(s)            | User                                                                                                                                                                                                                                                           |
| Precondition(s)     | The user has not previously registered in the system.                                                                                                                                                                                                          |
| Post-condition(s)   | A new account is created and the user is redirected to the Sign In page.                                                                                                                                                                                       |
| Main Flow(s)        | <ol style="list-style-type: none"> <li>1. User clicks the "Sign Up" button.</li> <li>2. User fills in registration details.</li> <li>3. System validates inputs.</li> <li>4. System creates a new user record.</li> <li>5. Redirects user to login.</li> </ol> |
| Alternative Flow(s) | –                                                                                                                                                                                                                                                              |
| Exception Flow(s)   | [E1] Email already exists → prompt user to try a different email.<br>[E2] Missing or invalid data → highlight errors and request correction.                                                                                                                   |

*Table 3.7: Use case explanation for the "Sign In" feature*

|                     |                                                                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name       | Sign In                                                                                                                                                                        |
| Description         | Authenticates existing users to allow access to system features.                                                                                                               |
| Actor(s)            | User                                                                                                                                                                           |
| Precondition(s)     | User has already registered.                                                                                                                                                   |
| Post-condition(s)   | User is granted access to the main system dashboard.                                                                                                                           |
| Main Flow(s)        | <ol style="list-style-type: none"> <li>1. User enters username and password.</li> <li>2. System checks credentials.</li> <li>3. System redirects user to dashboard.</li> </ol> |
| Alternative Flow(s) | –                                                                                                                                                                              |
| Exception Flow(s)   | [E1] User clicks "Sign Up" instead → redirect to registration page.<br>[E2] Incorrect credentials → display error and prompt to re-enter.                                      |

*Table 3.8: Use case explanation for the "Browse book" feature*

|               |                                                                   |
|---------------|-------------------------------------------------------------------|
| Use Case Name | Browse Book                                                       |
| Description   | Enables users to explore the catalog through filters or keywords. |

|                     |                                                                                                                                   |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Actor(s)            | User                                                                                                                              |
| Precondition(s)     | User must be signed in.                                                                                                           |
| Post-condition(s)   | List of books displayed based on filters/search.                                                                                  |
| Main Flow(s)        | 1. User navigates to the browse page.<br>2. Enters keywords or applies filters.<br>3. System fetches and displays matching books. |
| Alternative Flow(s) | –                                                                                                                                 |
| Exception Flow(s)   | [E1] No books found → show "No results" message.                                                                                  |

Table 3.9: Use case explanation for the "Borrow/Recommended Books" feature

|                     |                                                                                                                                 |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name       | Borrow / Recommended Books                                                                                                      |
| Description         | Users can borrow books or save them to bookmarks, especially from the recommendations.                                          |
| Actor(s)            | User                                                                                                                            |
| Precondition(s)     | User is signed in and has access to book catalogue.                                                                             |
| Post-condition(s)   | Book is marked as borrowed or bookmarked.                                                                                       |
| Main Flow(s)        | 1. User selects a book.<br>2. Chooses to "Borrow" or "Bookmark".<br>3. System verifies availability.<br>4. Updates book status. |
| Alternative Flow(s) | [A1] User chooses to bookmark instead of borrow.                                                                                |
| Exception Flow(s)   | [E1] Book not available → System offers "Notify Me" option.                                                                     |

Table 3.10: Use case explanation for the "View Recommended Books" feature

|                     |                                                                                                                 |
|---------------------|-----------------------------------------------------------------------------------------------------------------|
| Use Case Name       | View Recommended Books                                                                                          |
| Description         | Displays personalized or trend-based recommendations.                                                           |
| Actor(s)            | User                                                                                                            |
| Precondition(s)     | User has login access and borrowing/search history.                                                             |
| Post-condition(s)   | List of recommended books is shown.                                                                             |
| Main Flow(s)        | 1. User accesses recommendation section.<br>2. System fetches recommendations.<br>3. Displays suggested titles. |
| Alternative Flow(s) | –                                                                                                               |
| Exception Flow(s)   | [E1] No data found → Show generic trending books.                                                               |

Table 3.11: Use case explanation for the "Return Physical Book" feature

|                     |                                                                                                             |
|---------------------|-------------------------------------------------------------------------------------------------------------|
| Use Case Name       | Return Physical Book                                                                                        |
| Description         | Allows the user to return a previously borrowed physical book.                                              |
| Actor(s)            | User                                                                                                        |
| Precondition(s)     | Book has been borrowed.                                                                                     |
| Post-condition(s)   | Book return is logged and availability updated.                                                             |
| Main Flow(s)        | 1. User views borrowed books.<br>2. Clicks "Return Book".<br>3. Admin logs return and updates availability. |
| Alternative Flow(s) | –                                                                                                           |
| Exception Flow(s)   | [E1] Return fails → notify user to retry or contact admin.                                                  |

Table 3.12: Use case explanation for the "View Dashboard" feature

|                     |                                                                                     |
|---------------------|-------------------------------------------------------------------------------------|
| Use Case Name       | View Dashboard                                                                      |
| Description         | Admin monitors system status, usage statistics, and activities.                     |
| Actor(s)            | Admin                                                                               |
| Precondition(s)     | Admin is logged in.                                                                 |
| Post-condition(s)   | Dashboard data is displayed.                                                        |
| Main Flow(s)        | 1. Admin logs in.<br>2. Navigates to dashboard.<br>3. Dashboard loads current data. |
| Alternative Flow(s) | –                                                                                   |
| Exception Flow(s)   | [E1] No data available → show empty placeholders.                                   |

Table 3.13: Use case explanation for the "Add/Edit Book" feature

|                     |                                                                                                         |
|---------------------|---------------------------------------------------------------------------------------------------------|
| Use Case Name       | Add/Edit Book                                                                                           |
| Description         | Admin can add new books or update book information.                                                     |
| Actor(s)            | Admin                                                                                                   |
| Precondition(s)     | Admin is authenticated.                                                                                 |
| Post-condition(s)   | Book catalogue is updated.                                                                              |
| Main Flow(s)        | 1. Admin accesses book list.<br>2. Enters or edits book data.<br>3. System validates and saves changes. |
| Alternative Flow(s) | –                                                                                                       |
| Exception Flow(s)   | [E1] Invalid data → display error for correction.                                                       |

Table 3.14: Use case explanation for the "Manage Requests" feature

|                     |                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Use Case Name       | Manage Requests                                                                                 |
| Description         | Admin reviews and processes user borrowing requests.                                            |
| Actor(s)            | Admin                                                                                           |
| Precondition(s)     | Requests have been submitted.                                                                   |
| Post-condition(s)   | Requests are approved or rejected.                                                              |
| Main Flow(s)        | 1. Admin checks request list.<br>2. Clicks approve/reject.<br>3. System updates request status. |
| Alternative Flow(s) | –                                                                                               |
| Exception Flow(s)   | [E1] Request expired → system shows as invalid.                                                 |

Table 3.15: Use case explanation for the "Notify Genre" feature

|                     |                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------|
| Use Case Name       | Notify New Genre                                                                                                  |
| Description         | System sends alerts to users when books of interest are added.                                                    |
| Actor(s)            | System                                                                                                            |
| Precondition(s)     | Admin adds a new book that matches user preferences.                                                              |
| Post-condition(s)   | Users are notified about the new book.                                                                            |
| Main Flow(s)        | 1. Admin adds a new book.<br>2. System checks user genre preferences.<br>3. Sends notification to relevant users. |
| Alternative Flow(s) | –                                                                                                                 |
| Exception Flow(s)   | [E1] No matching users → no notifications sent.                                                                   |

### 3.2.2.2 Activity Diagram

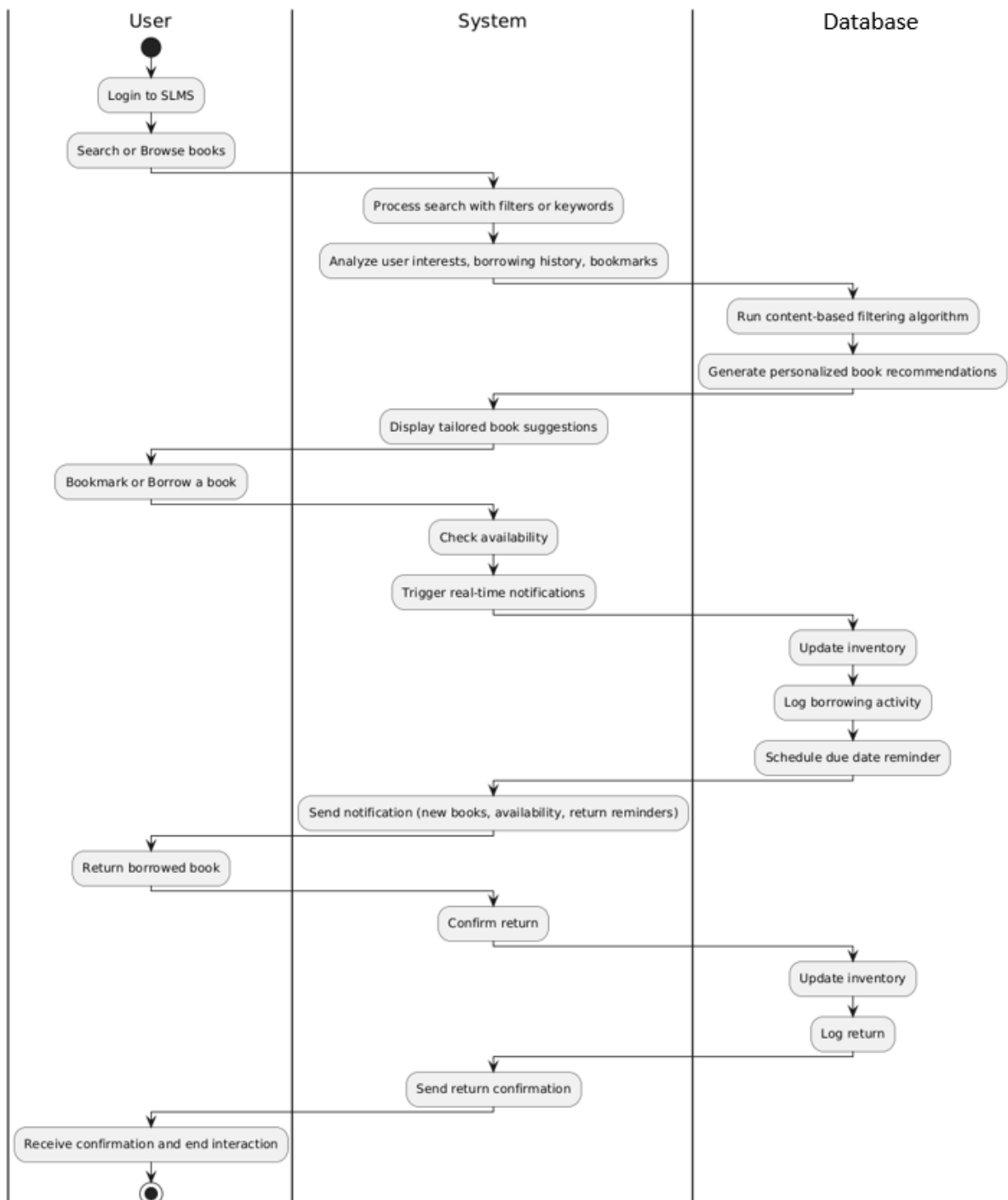


Figure 3.2: Activity Diagram for User

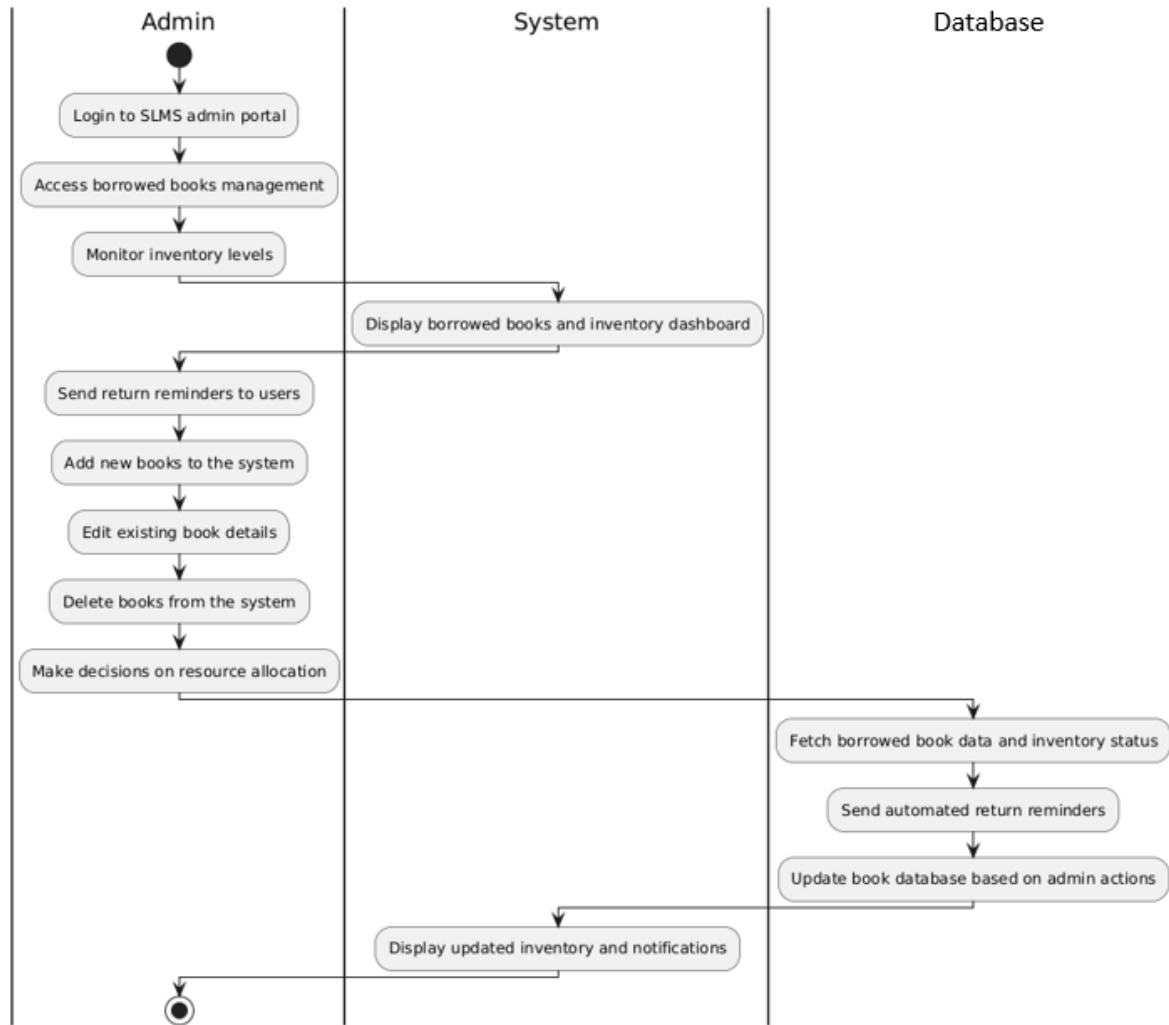


Figure 3.3: Activity Diagram for Admin

Figure 3.2 and 3.3 illustrates the activity diagram of the proposed system for user and admin. Activity diagrams are behavioral diagrams that represent the flow of control from one activity to another within a system. They capture the dynamic aspects by showing the sequence of activities starting from an initial point through various possible paths until the process completes (Fowler, 2003).

### 3.2.2.3 Sequence Diagram

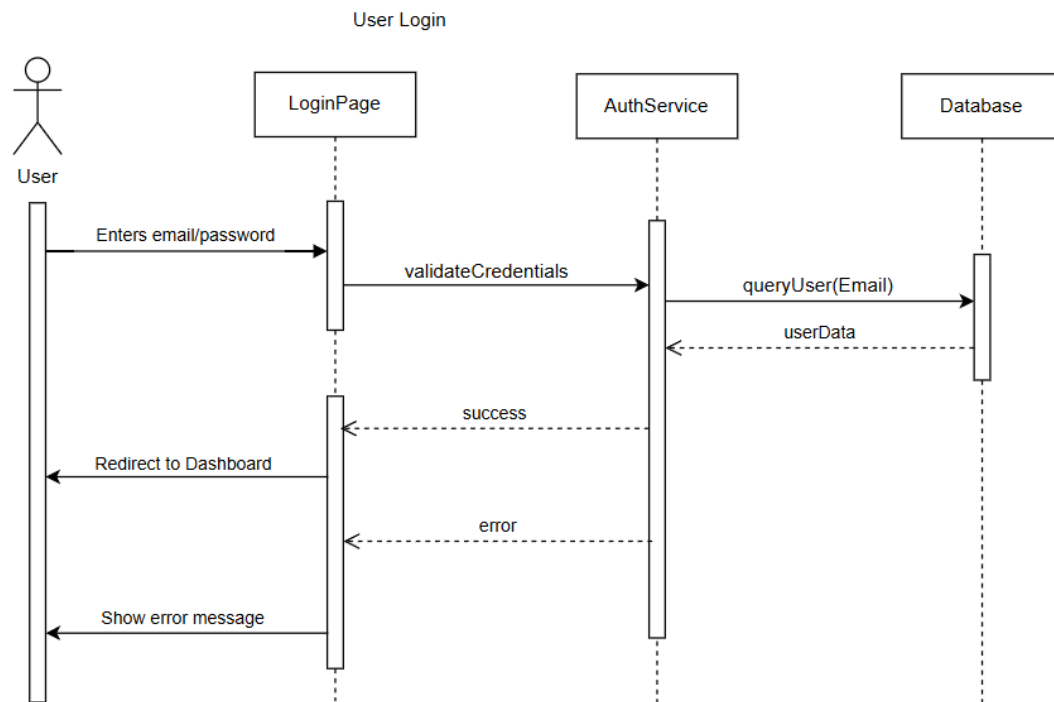


Figure 3.4: Sequence Diagram for User Login

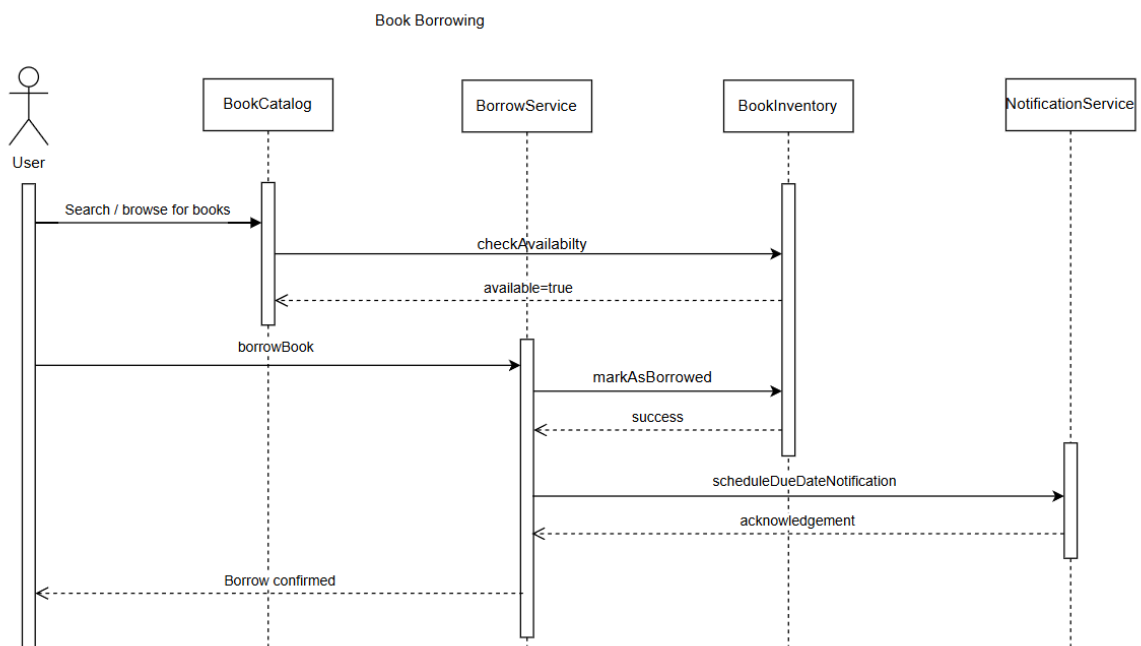


Figure 3.5: Sequence Diagram for Book Borrowing

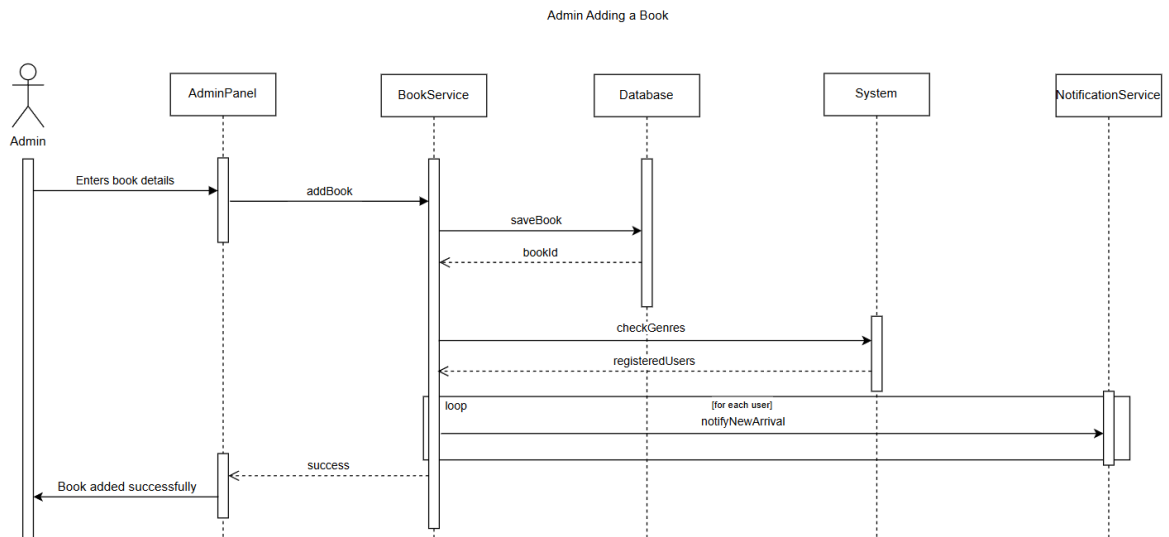


Figure 3.6: Sequence Diagram for Admin Adding a Book

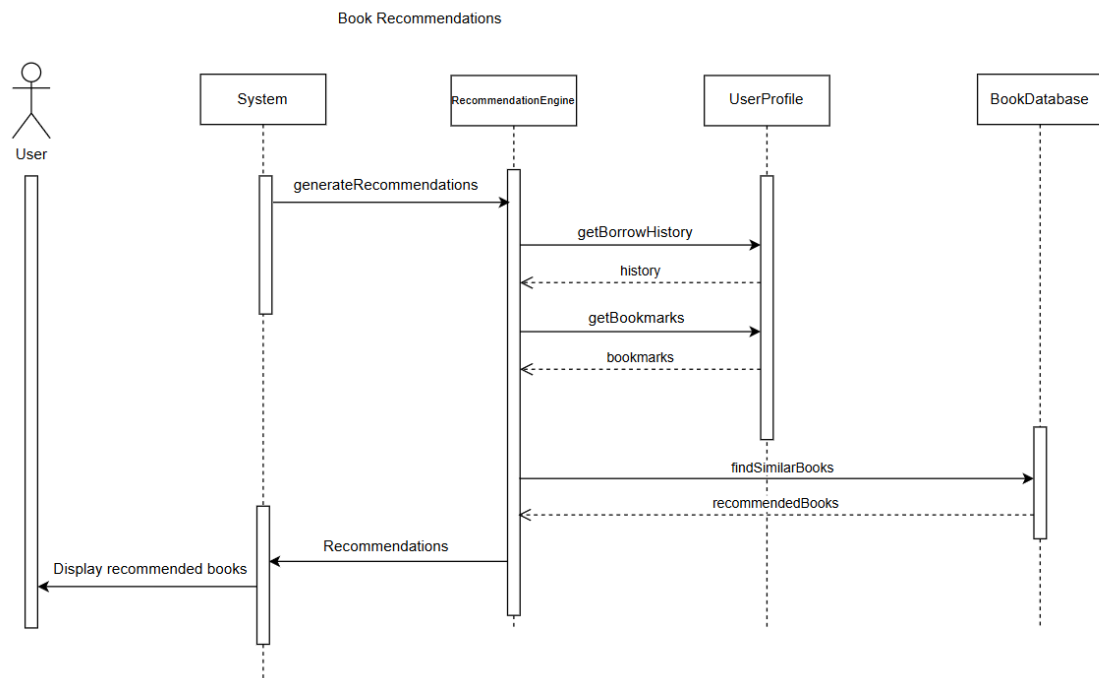
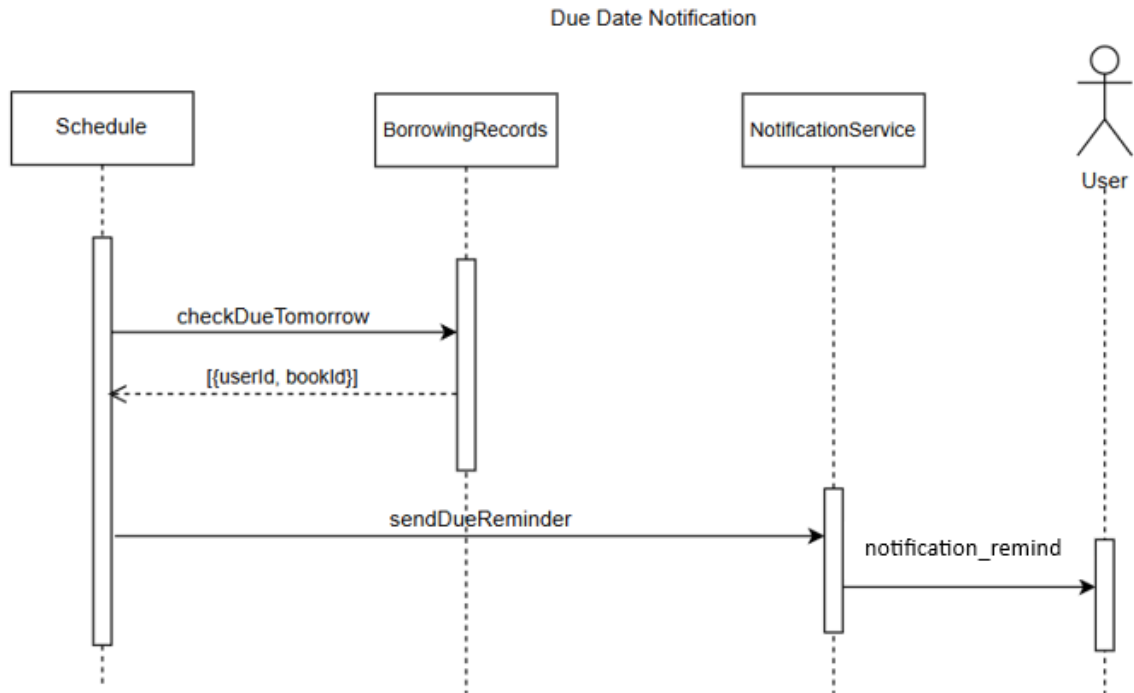


Figure 3.7: Sequence Diagram for Book Recommendation



*Figure 3.8: Sequence Diagram for Due Date Notification*

Figures 3.4 to 3.8 present sequence diagrams that illustrate various interactions within the proposed system. A sequence diagram is a type of interaction diagram that details how system operations are carried out step-by-step. Specifically, Figure 3.4 shows the user login process, Figure 3.5 illustrates the book borrowing procedure, and Figure 3.6 depicts the admin adding a book to the system. Additionally, Figure 3.7 demonstrates how personalized book recommendations are generated, while Figure 3.8 shows the notification process for due dates.

### 3.2.2.4 Class Diagram

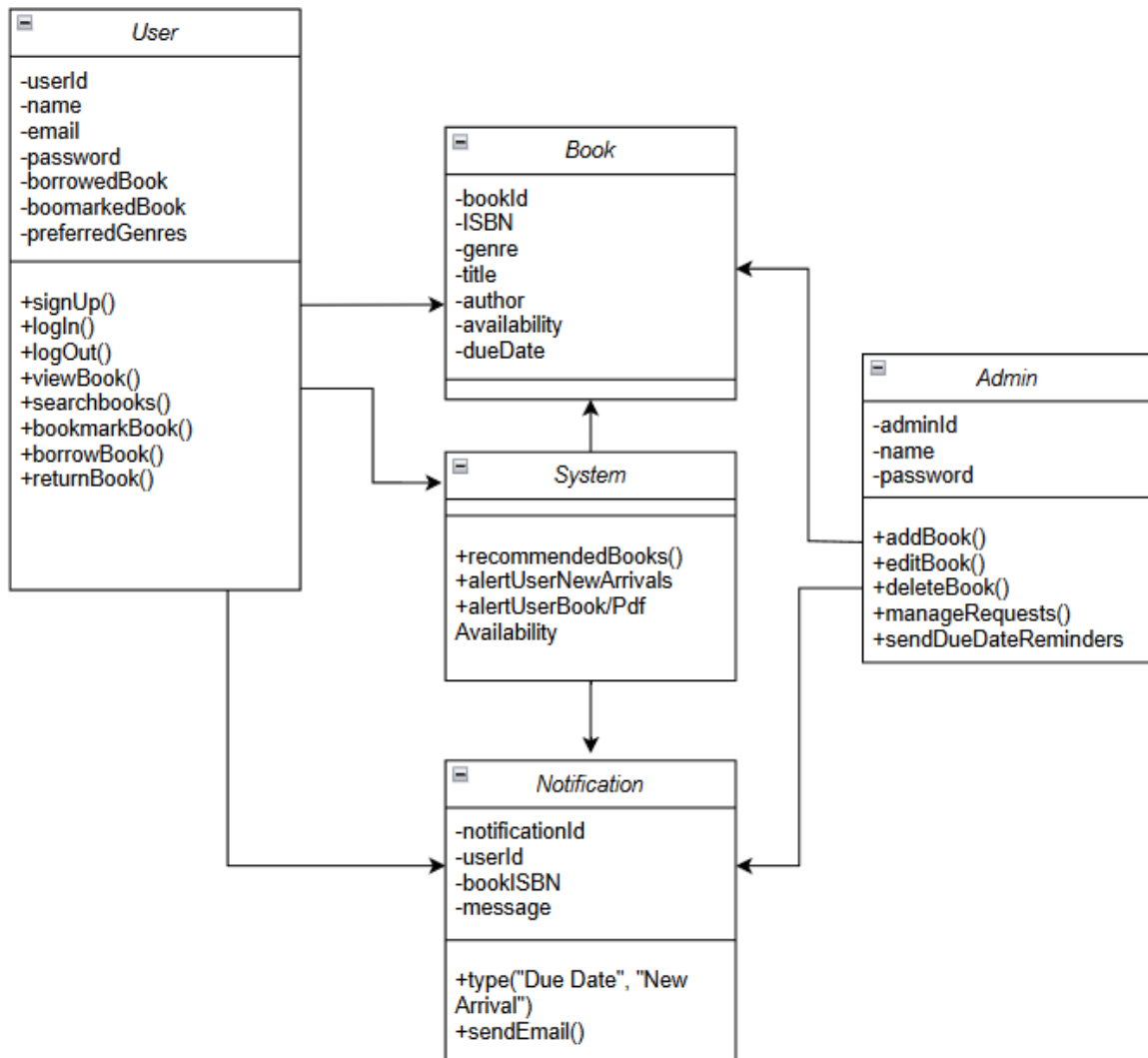


Figure 3.9: Class Diagram

Figure 3.9 shows the main parts of the Smart Library Management System and how they work together. The system has Users who can search for, borrow, and bookmark books. Each user has a profile with their details and a list of books they have borrowed or saved. Users also receive notifications about things like due dates or new books.

The admin is the librarian who manages the library's collection by adding, editing, or deleting books. The admin also sends reminders to users about when borrowed books need to be returned.

The Book class represents the books in the library. Each book has information like its title, author, genre, whether it is available, and its due date if it is borrowed.

The Notification class handles messages sent to users, such as reminders about due dates or announcements about new books. Notifications contain details like the message content and the type of notification.

The System class runs important functions like recommending books to users based on their interests and borrowing history. It also alerts users about new books in their favourite genres.

In short, users interact with the system to find and borrow books, while admins manage the book collection and send notifications. The system helps by recommending books and sending timely alerts, making the library experience smooth and personalized for everyone.

### 3.2.2.5 Interface Design

Interface design is used to plan and visualize the design of the proposed system, focusing on user interaction and overall experience. The main page of the system will begin with a Sign In page, as shown in Figure 3.10.



**Welcome Back !!**

Please enter your credentials to log in

Username

Password

[Forgot password?](#)  
[Sign Up](#)

**SIGN IN**

*Figure 3.10: Sign-in Page*

To access the system, users must enter their registered username and password on the sign-in page. For new users who have not yet registered, they can do so by clicking the 'Sign Up' button. This action will redirect them to the Sign-Up page, where they are required to provide their username, email address, password and confirm password to complete the registration process, as illustrated in Figure 3.11.



Smart Library  
Management  
System

### Sign Up

Username

Email Address

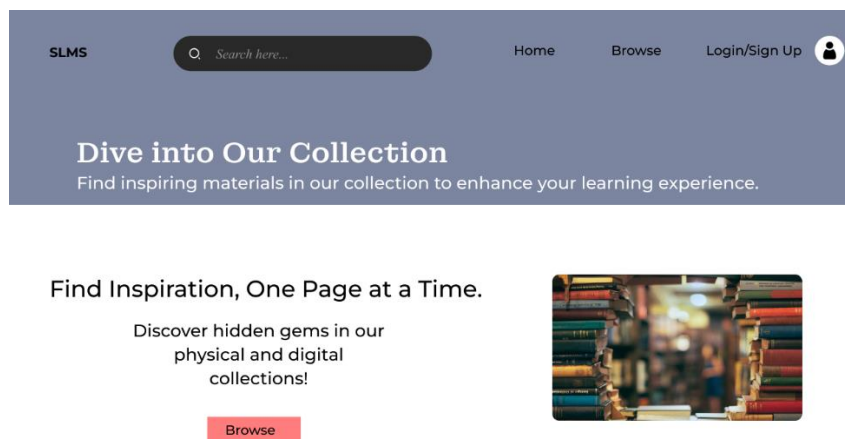
Password

Confirm Password

Sign Up

*Figure 3.11: Sign Up Page*

After completing the registration process, the user must sign in using the registered username and password. Upon successful login, the system will direct the user to the homepage, as shown in Figure 3.12. On the homepage, the user will find several key features, including a search button, home button, browse button, login/signup button, and a profile icon.



*Figure 3.12: Homepage*

When the user clicks on the Browse button, they will be directed to the Browse Page, as shown in Figure 3.13. On this page, users can explore and discover books they are interested in. Additionally, the system will display recommended books based on the user's past bookmarking and borrowing history. As illustrated in Figure 3.14, users can also search for books and apply filters such as genre, publication year, and availability for borrowing to refine their search results.

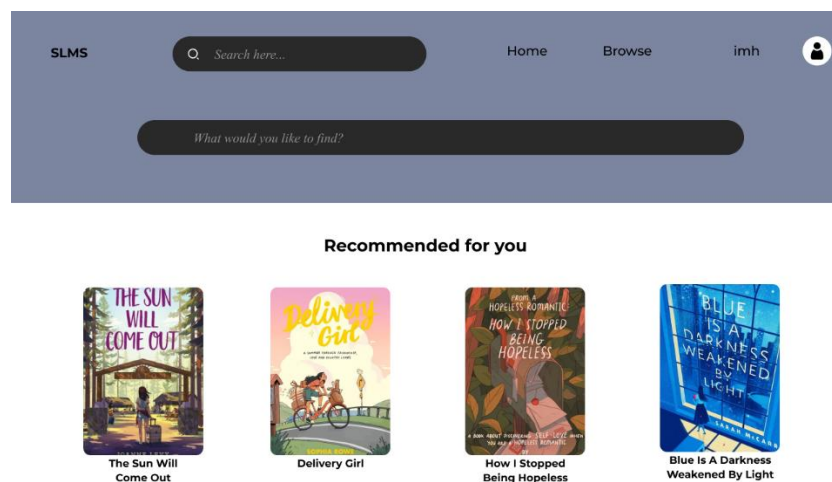


Figure 3.13: Recommended Book for User

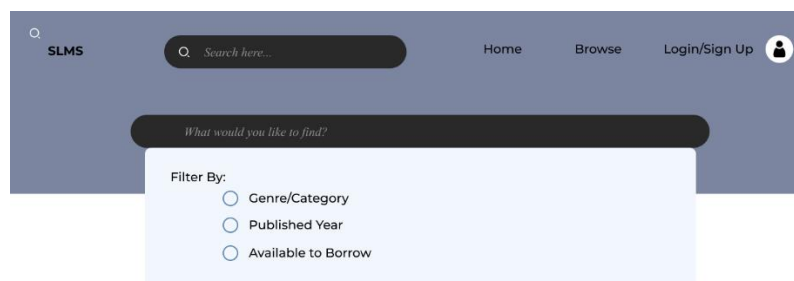
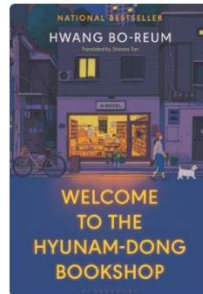


Figure 3.0.14: Filter Books

Next, when the user clicks on a book to view more details, they will be directed to the View Book page, as shown in Figure 3.15. This page displays important information such as the book title, a brief synopsis, and action buttons including Borrow, Read, and Bookmark. The user can choose any of these options based on their preference whether they want to borrow the book, read it online, or bookmark it for future reference.



### Welcome To The Hyunam-Dong Bookshop



This book is available on shelf at **SLMS**

Borrow

Read

#### Synopsis

Yeongju is burned out. With her high-flying career, demanding marriage, and bustling life in Seoul, she knows she should feel successful—but all she feels is drained. Haunted by an abandoned dream, she takes a leap of faith and leaves her old life behind. Quitting her job and divorcing her husband, Yeongju moves to a quiet residential neighborhood outside the city and opens the Hyunam-dong Bookshop.

The transition isn't easy. For months, all Yeongju can do is cry. But as the long hours in the shop stretch on, she begins to reflect on what makes a good bookseller and a meaningful store. She throws herself into reading voraciously, hosting author events, and crafting her own philosophy on bookselling. Gradually, Yeongju finds her footing in her new surroundings.

Surrounded by friends, writers, and the books that bind them, Yeongju begins to write a new chapter in her life. The Hyunam-dong Bookshop evolves into a warm, welcoming haven for lost souls—a place to rest, heal, and remember that it's never too late to scrap the plot and start over.

Figure 3.15: View Book Page

Next, the user can click on the Notification icon to view important updates, such as due date reminders, notifications for accepted or rejected book borrowing requests, and a list of books currently borrowed. These details are displayed on the Notification Page, as shown in Figure 3.16.

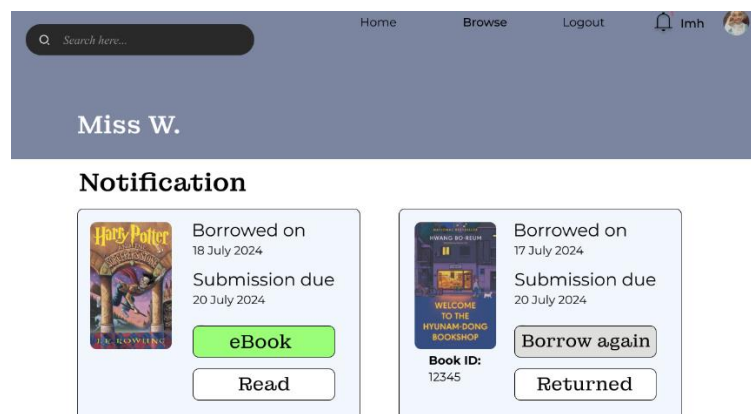


Figure 3.16: Notification Page

For the admin, after logging into the system, they will be directed to the Dashboard, as shown in Figure 3.17. On this page, the admin can view key statistics such as the total number of registered books, borrowed books, genres, and user requests. The dashboard also includes several action buttons for administrative tasks, including Manage Books, Book List, and Manage Requests.

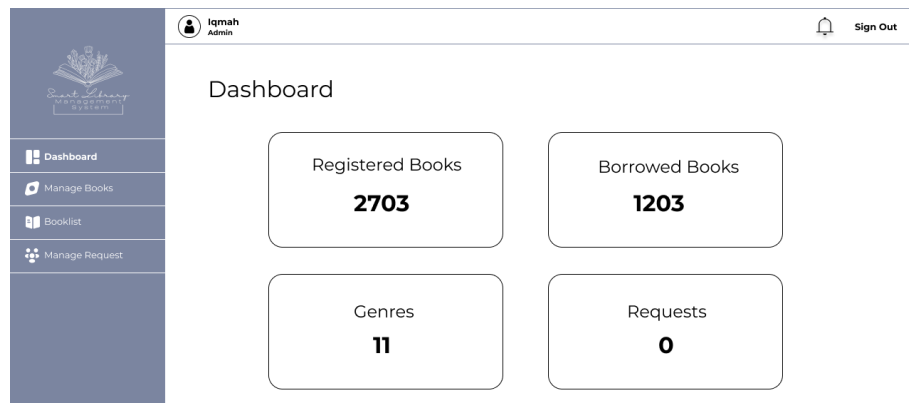


Figure 3.17: Dashboard

Next, when the admin clicks on Manage Books, they will be able to view a list of books that are currently borrowed by users, as shown in Figure 3.18. This section displays important details such as the book title, the user who borrowed it, and the corresponding due date.

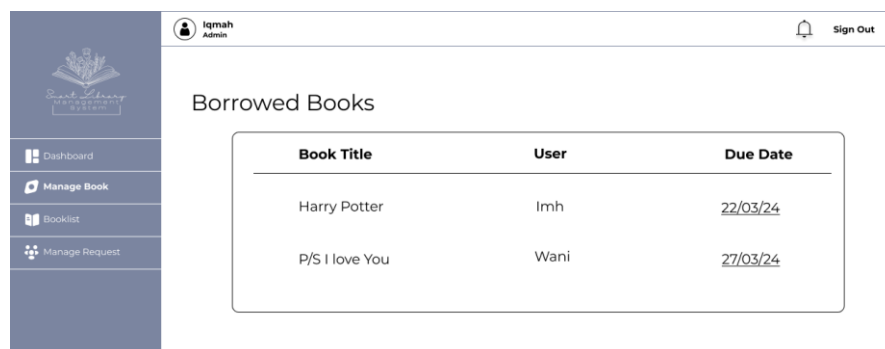


Figure 3.18: Manage Book

When the admin clicks on the Book List button, they will be presented with a complete list of books, as shown in Figure 3.19. This list includes key information such as the Book ID, Title, Genre, Language, and Availability status. On this page, the admin also has the options to edit, delete, or view detailed information for each book.



 Dashboard

 Manage Book

 Booklist

 Manage Request

 Iqmah Admin

 Sign Out

Book Management

 Add Book

| ID | Name             | Genre   | Language | Availability | Action                                                                                                                                                                                                                                                      |
|----|------------------|---------|----------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Once Upon a Time | Fiction | English  | Borrowed     |    |
| 1  | Once Upon a Time | Fiction | English  | Available    |    |
| 1  | Once Upon a Time | Fiction | English  | Borrowed     |    |
| 1  | Once Upon a Time | Fiction | English  | Available    |    |
| 1  | Once Upon a Time | Fiction | English  | Available    |    |
| 1  | Once Upon a Time | Fiction | English  | Borrowed     |    |
| 1  | Once Upon a Time | Fiction | English  | Available    |    |
| 1  | Once Upon a Time | Fiction | English  | Borrowed     |    |
| 1  | Once Upon a Time | Fiction | English  | Available    |    |

Figure 3.19: Booklist

Lastly, under the Manage Requests section, as shown in Figure 3.20, the admin can view a list of books borrowing requests submitted by users. For each request, the admin has the option to either accept or reject it based on availability and other relevant criteria.



 Dashboard

 Manage Book

 Booklist

 Manage Requests

 Iqmah Admin

 Sign Out

# Request

| ID | User | Email            | Username | Request | Action        |
|----|------|------------------|----------|---------|---------------|
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |
| 1  | Wani | kmvshi@gmail.com | kmvshi   | Borrow  | Accept/Reject |

Figure 3.20: Manage Requests

### **3.2.2.6 How does the algorithm works?**

The SLMS's recommendation algorithm works using genre-based filtering focused specifically on book genres. It tracks the genres of books that a user has previously borrowed or bookmarked and identifies which genres the user prefers most. Based on this data, the system suggests other books within those same genres, ensuring recommendations are aligned with the user's interests. As the user interacts more with the system, borrowing or bookmarking additional books, the algorithm updates the preferred genres accordingly to keep recommendations relevant and personalized over time. This genre-focused approach helps users discover books they are most likely to enjoy without overwhelming them with unrelated options.

### **3.2.3 Implementation**

The SLMS is implemented using VS Code as the primary development environment because of its versatility and powerful features for web development. VS Code supports multiple programming languages and offers helpful tools like syntax highlighting, code completion, and integrated debugging, which streamline the coding process. The system's backend is developed using PHP, a popular server-side scripting language, which handles tasks such as processing user requests, managing sessions, and interacting with the database to store and retrieve data. For the front-end design, CSS is used to create an attractive and responsive user interface that ensures a smooth and user-friendly experience across different devices. JavaScript is incorporated to enhance the interactivity of the system by enabling dynamic content updates without the need to reload pages.

To run and test the system locally during development, XAMPP is employed. XAMPP is an easy-to-install software package that provides an Apache web server and a MySQL database server on the developer's computer. This setup mimics a real web server environment, allowing developers to test how the system will behave in production

without the need for an internet connection or an external server. The development workflow involves writing code in VS Code, running the system through the XAMPP server, and continuously testing and debugging the application. Errors and bugs can be quickly identified and resolved using the debugging tools available in both VS Code and XAMPP's control panel. This approach ensures that the SLMS is thoroughly tested and refined before it is deployed to a live environment.

### **3.2.4 Testing**

Following the implementation phase, the next step is testing, which aims to verify that all system functions operate as intended according to the initial design specifications. This phase focuses on assessing both the functionality and usability of the SLMS to ensure it is free of errors and performs smoothly.

The system will be tested with a selected group of participants who represent the target audience. Their interactions with the system will be observed to evaluate how intuitively they can navigate and use the various features.

Usability testing will play a key role in this phase by measuring how effectively users can accomplish tasks and how easily they understand the interface. Feedback will be gathered to identify any usability issues, misunderstandings, or difficulties encountered by users. Additionally, the accuracy and relevance of the book recommendations generated by the system will be reviewed to ensure that the personalized suggestions meet user expectations.

Overall, this testing phase is critical for uncovering potential issues and gathering valuable user insights that will guide final adjustments, resulting in a more reliable and user-friendly system.

### **3.2.5 Maintenance**

After deployment, SLMS will undergo regular maintenance to ensure continued reliability, security, and performance. Maintenance activities include fixing any bugs or errors that arise, updating the system to address new user requirements, and improving features based on user feedback. Routine checks will also be performed to ensure the database remains optimized and secure. Additionally, periodic updates to the software components, such as PHP, JavaScript libraries, and server environments, will be carried out to maintain compatibility and protect against vulnerabilities. Effective maintenance is essential to keep the system running smoothly and to adapt to evolving user needs over time.

### **3.3 Summary**

This chapter provides a detailed overview of the methodology used in developing the Smart Library Management System. It begins with the requirement analysis phase, covering both system and user requirements to establish the foundation for the project. The system design section outlines the planning and architectural decisions that guide the development process. Following the design, the implementation phase describes the tools and technologies used to build the system. The testing phase explains how the system's functionality and usability are evaluated to ensure it meets the project goals. Finally, the maintenance section highlights the ongoing support and updates needed to keep the system reliable and effective. Overall, this chapter presents a comprehensive view of the development lifecycle for the proposed system.

## **CHAPTER 4**

### **IMPLEMENTATION**

#### **4.1 Overview**

This chapter documents the development and implementation of the SLMS. The proposed solution and system design outlined earlier in the chapter serve as the foundation for building the system. During the development phase, all necessary components, tools, and software such as VS Code, XAMPP, PHP, CSS, and JavaScript are set up to begin the actual system construction.

#### **4.2 Installation and System Configuration**

Setting up the SLMS requires the installation and configuration of specific tools to support its development and operation. VS Code is used as the integrated development environment (IDE), while XAMPP serves as the local server environment, offering both Apache and MySQL functionalities. These tools are essential for running and testing the system on a local machine. All development and testing activities are conducted using a standard laptop, which functions as the primary hardware platform.

##### **4.2.1 Software Installation Procedure**

Before starting development, the required software must be installed in the correct order to ensure the SLMS operates efficiently. The software tools include XAMPP and Visual Studio Code, both of which are essential for creating, testing, and running the system.

##### **4.2.1.1 XAMPP Installation**

XAMPP is an all-in-one software package that provides a local web server environment for PHP applications. It includes the Apache server and MySQL database, which are critical components for running the SLMS on a local machine. To begin, XAMPP

is downloaded and installed from its official website. Once installed, users must open the XAMPP Control Panel and start both the Apache and MySQL modules. This step ensures that the system has an active server and database environment necessary for SLMS to function properly during development and testing.

#### **4.2.1.2 VS Code Installation**

VS Code is used as the primary code editor for developing the SLMS. It supports various extensions and features that enhance coding productivity and make it easier to work with languages such as PHP, JavaScript, and CSS. After installing VS Code, a dedicated project folder for SLMS is created within the htdocs directory of the XAMPP installation. This directory serves as the root folder where all project files, including backend scripts and frontend components, are stored.

#### **4.2.1.3 Database Setup Using phpMyAdmin**

The MySQL database for SLMS is configured using phpMyAdmin, a web-based database management tool included with XAMPP. Through phpMyAdmin, a new database is created to store all system data. The database contains tables such as acceptedbooks, authors, bookauthors, bookcounts, bookgenres, bookmarks, bookrequests, books, genres, notifications, notify\_requests, and users, as shown in Figure 4.1. These tables are designed according to the database schema to support essential system functions, including book management, user authentication, and handling of borrowing requests.

| Table            | Action                                    | Rows       | Type          | Collation                 | Size             | Overhead   |
|------------------|-------------------------------------------|------------|---------------|---------------------------|------------------|------------|
| acceptedbooks    | Browse Structure Search Insert Empty Drop | 1          | InnoDB        | utf8mb4_general_ci        | 32.0 KiB         | -          |
| authors          | Browse Structure Search Insert Empty Drop | 48         | InnoDB        | utf8mb4_general_ci        | 16.0 KiB         | -          |
| bookauthors      | Browse Structure Search Insert Empty Drop | 24         | InnoDB        | utf8mb4_general_ci        | 32.0 KiB         | -          |
| bookcounts       | Browse Structure Search Insert Empty Drop | 1          | InnoDB        | utf8mb4_general_ci        | 16.0 KiB         | -          |
| bookgenres       | Browse Structure Search Insert Empty Drop | 24         | InnoDB        | utf8mb4_general_ci        | 32.0 KiB         | -          |
| bookmarks        | Browse Structure Search Insert Empty Drop | 0          | InnoDB        | utf8mb4_general_ci        | 48.0 KiB         | -          |
| bookrequests     | Browse Structure Search Insert Empty Drop | 4          | InnoDB        | utf8mb4_general_ci        | 48.0 KiB         | -          |
| books            | Browse Structure Search Insert Empty Drop | 24         | InnoDB        | utf8mb4_general_ci        | 16.0 KiB         | -          |
| genres           | Browse Structure Search Insert Empty Drop | 11         | InnoDB        | utf8mb4_general_ci        | 16.0 KiB         | -          |
| notifications    | Browse Structure Search Insert Empty Drop | 13         | InnoDB        | utf8mb4_general_ci        | 48.0 KiB         | -          |
| notify_requests  | Browse Structure Search Insert Empty Drop | 2          | InnoDB        | utf8mb4_general_ci        | 16.0 KiB         | -          |
| users            | Browse Structure Search Insert Empty Drop | 3          | InnoDB        | utf8mb4_general_ci        | 48.0 KiB         | -          |
| <b>12 tables</b> | <b>Sum</b>                                | <b>155</b> | <b>InnoDB</b> | <b>utf8mb4_general_ci</b> | <b>368.0 KiB</b> | <b>0 B</b> |

Figure 4.1: Librarywebsite Table

#### 4.2.1.4 Running the System Locally

Once the server, code editor, and database are properly configured, the SLMS can be launched in a local development environment. This is done by opening a web browser and navigating to <http://localhost/library> as shown in Figure 4.2. From there, developers can begin testing and interacting with the system's features, ensuring it runs as expected during development.

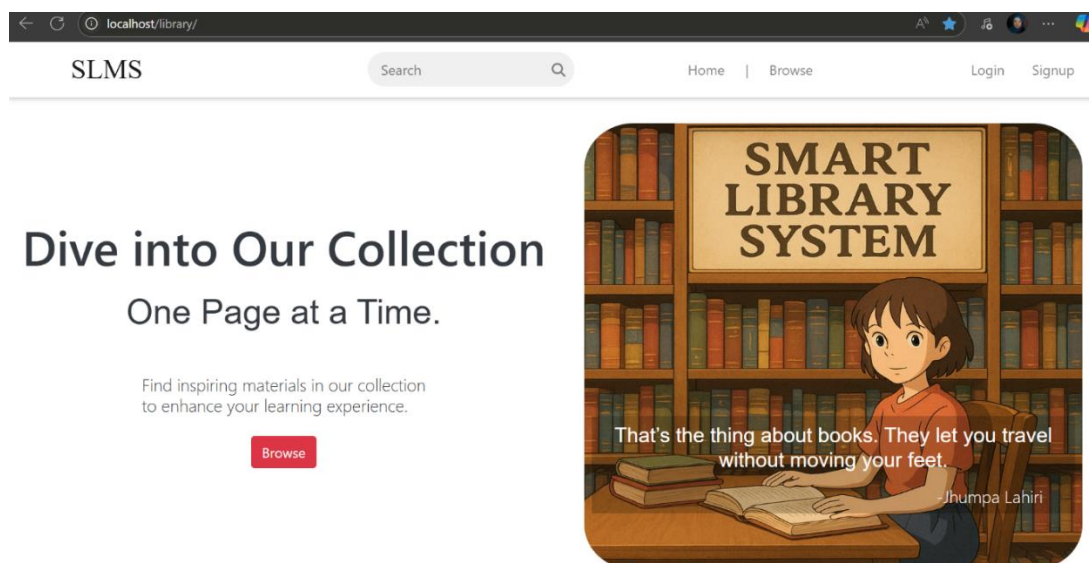
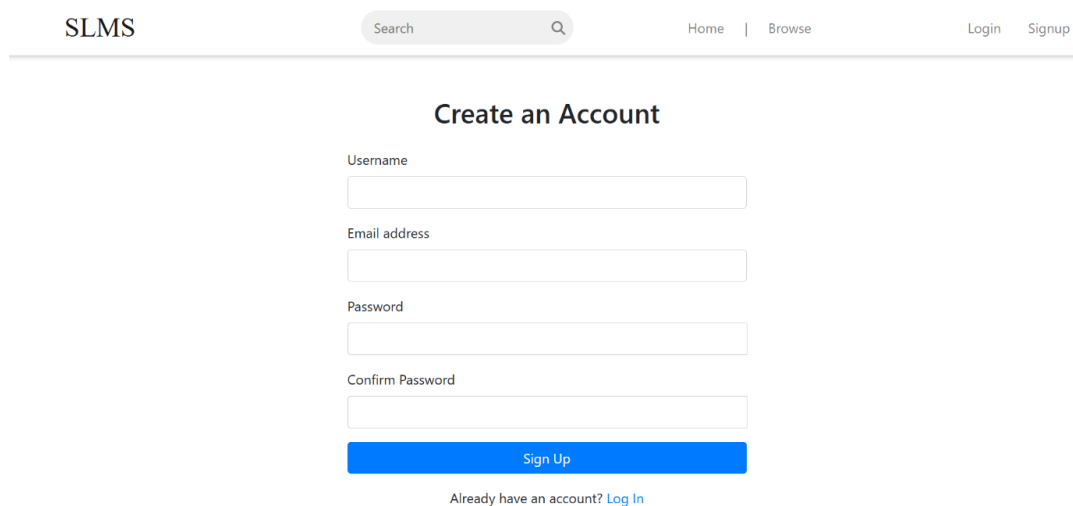


Figure 4.2: Smart Library Management System

## 4.3 Functions of the System

### 4.3.1 Sign up Page

The Sign-up Page lets new users create an account by entering their username, email address, and password, as shown in Figure 4.3. The system checks that the email is in the correct format to keep data accurate and to send important messages like new book arrivals, due reminders, and book availability. After signing up successfully, users are taken to the Sign-in Page. The system also makes sure that usernames and email addresses are not already used, as shown in Figure 4.4.



The screenshot displays the 'Create an Account' form on the SLMS website. The header includes the 'SLMS' logo, a search bar, and navigation links for 'Home', 'Browse', 'Login', and 'Signup'. The form itself is centered and contains four input fields: 'Username', 'Email address', 'Password', and 'Confirm Password'. A blue 'Sign Up' button is positioned below the 'Confirm Password' field. At the bottom of the form, there is a link that reads 'Already have an account? Log In'.

*Figure 4.3: Sign up Page*

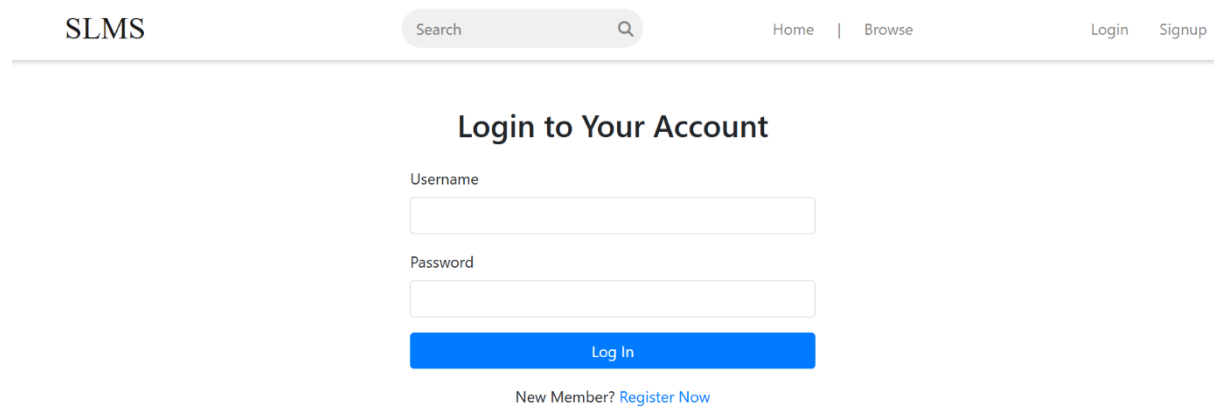
The screenshot shows the 'Create an Account' page of the SLMS system. At the top, there is a navigation bar with the SLMS logo, a search bar, and links for Home, Browse, Login, and Signup. A red banner at the top of the main content area displays the message 'Username or email already exists.' Below this, the 'Create an Account' heading is centered. The form consists of four input fields: Username, Email address, Password, and Confirm Password. A blue 'Sign Up' button is positioned below the form. At the bottom, a link 'Log In' is provided for users who already have an account.

*Figure 4.4: Username or Email Already Exists*

#### 4.3.2 Sign-in Page

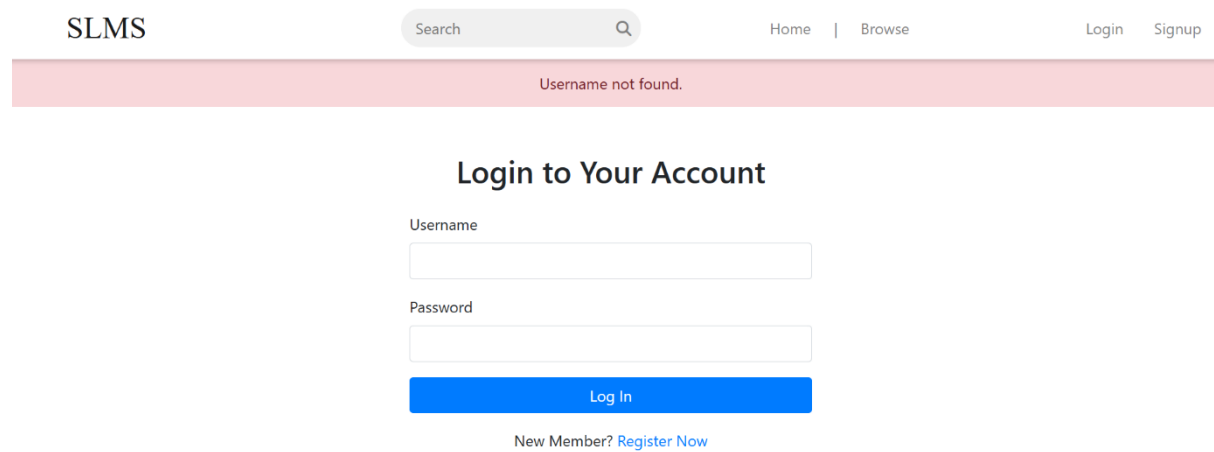
The Sign-in Page serves as the entry point for all registered users to access the system. Users must enter their correct username and password to log in, as shown in Figure 4.5. The system checks the input and displays a "Username not found" message if the entered username does not exist as shown in Figure 4.6. If the username exists but the password is incorrect, an "Incorrect password" message will be shown as in Figure 4.7. Upon successful login, the user will be directed to the main page. If the user does not have an account, they can click the register now link below the Log In button to create a new

one. Basic validation is included to ensure proper input format and to manage repeated login attempts.



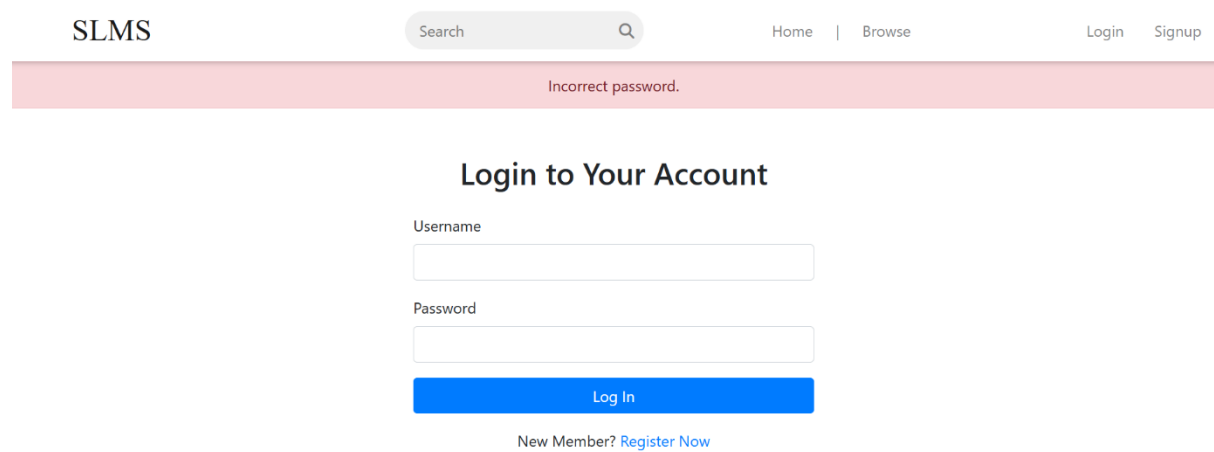
The image shows the 'Login to Your Account' page of the SLMS system. At the top is a navigation bar with the 'SLMS' logo, a search bar, and links for 'Home', 'Browse', 'Login', and 'Signup'. The main heading is 'Login to Your Account'. Below it are two input fields: 'Username' and 'Password'. A blue 'Log In' button is positioned below the password field. At the bottom of the form, there is a link that says 'New Member? Register Now'.

Figure 4.5: Sign in Page



This image shows the login page with an error message. A pink banner at the top of the main content area displays the text 'Username not found.' Below the banner, the 'Login to Your Account' form remains visible, including the 'Username' and 'Password' fields, the 'Log In' button, and the 'New Member? Register Now' link.

Figure 4.6: Error Username Not Found



This image shows the login page with an error message. A pink banner at the top of the main content area displays the text 'Incorrect password.' Below the banner, the 'Login to Your Account' form is visible, featuring the 'Username' and 'Password' input fields, the 'Log In' button, and the 'New Member? Register Now' link.

Figure 4.7: Error Incorrect Password

### 4.3.3 SLMS Homepage

After logging in, users are directed to the SLMS Homepage, which serves as the main hub for navigation (Figure 4.8). At the top, a navigation bar displays icons or labels for Home, Search, Browse, and Profile. The main page includes a Browse button, a Trending Books section (Figure 4.9), Inspirational Quotes (Figure 4.10), and a Recommended Books section where books are displayed randomly to offer variety (Figure 4.11). A footer is also included at the bottom of the page, providing additional information about the library (Figure 4.11). The homepage features a clean and user-friendly layout, making it easy for users to access the system's main features.

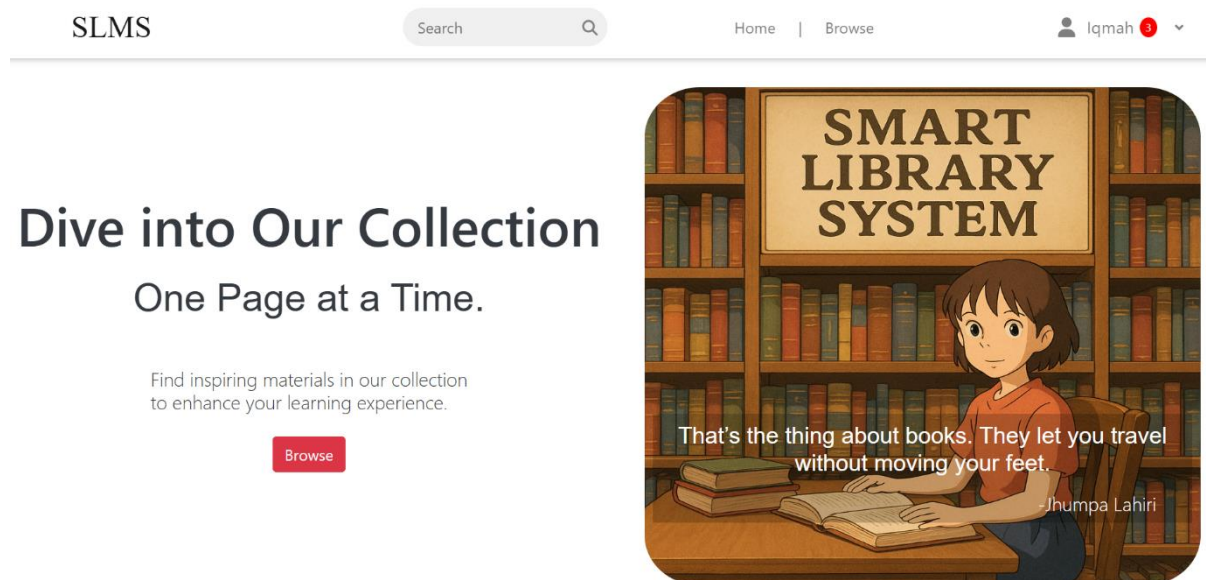


Figure 4.8: Homepage

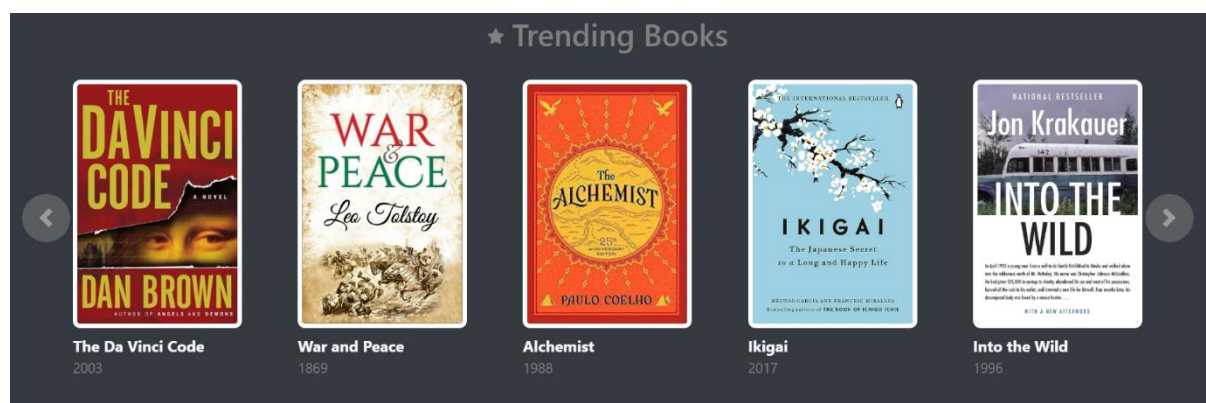


Figure 4.9: Trending Books

## Did you know?



- If you read 20 minutes a day, you would have read 1.8 million words in a year.
- Reading books can actually reduce stress levels.
- There's a word for the fear of running out of reading material. (Abibliophobia)
- There are around 130 million published books.
- The first ever story written was 'The Epic of Gilgamesh'.

Figure 4.10: Inspirational Quotes

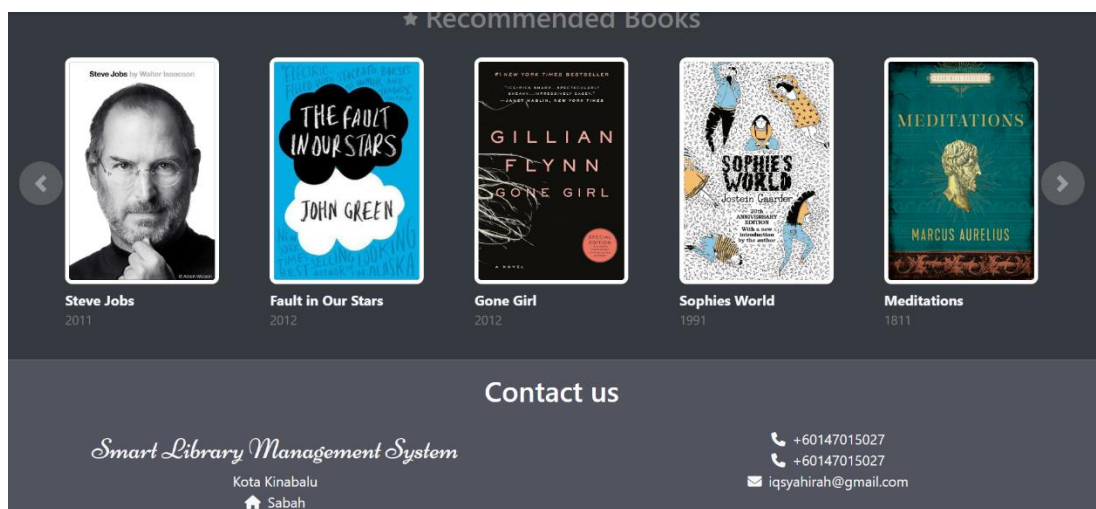


Figure 4.11: Recommended Books and Footer

### 4.3.4 Browse Book Page

The Browse Book Page offers users a comprehensive view of the library's available collection (Figure 4.12). Users can filter books by multiple criteria including genre, publication year, and borrowing availability, making it easier to find books that meet their preferences. This page dynamically updates recommendations based on the user's previous interactions, such as bookmarked or borrowed books. Users can click on any book to view more detailed information or initiate borrowing actions directly from this page.

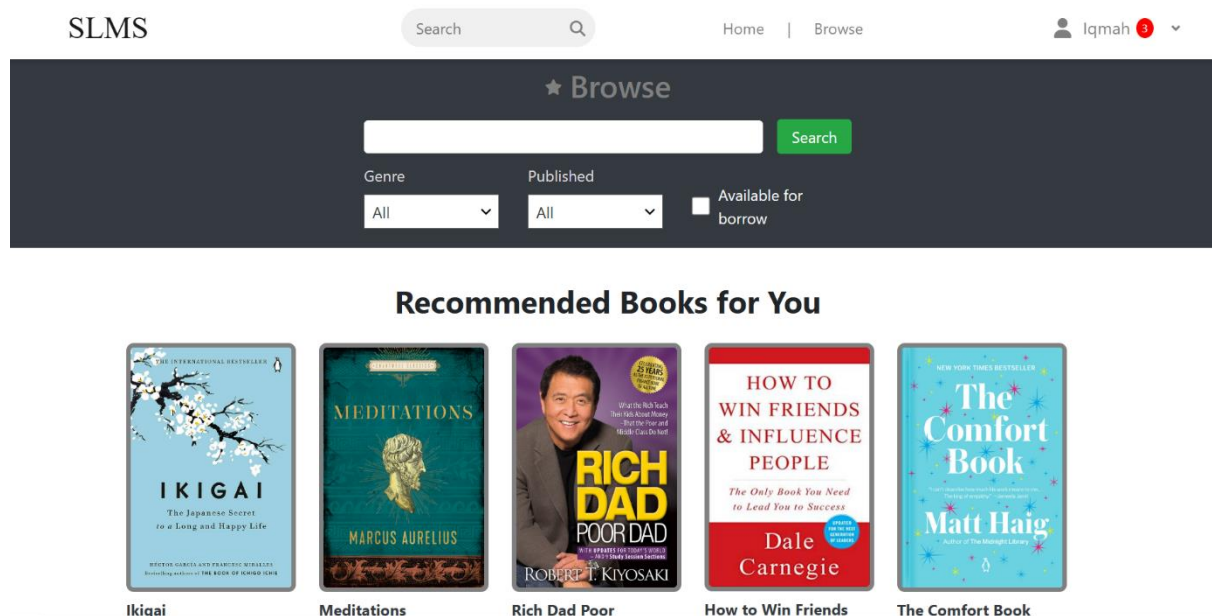


Figure 4.12: Browse Page

#### 4.3.5 View Book Page

This page displays detailed information about a selected book, including the title, author(s), synopsis, publication date, and genre, as shown in Figure 4.13. It also provides interactive options such as a Borrow button to request the book, a Read button to access digital copies if available, and a Bookmark button to save the book for future reference.

Additionally, a Related Books section is included to help users discover similar titles based on genre or theme, as shown in Figure 4.14.

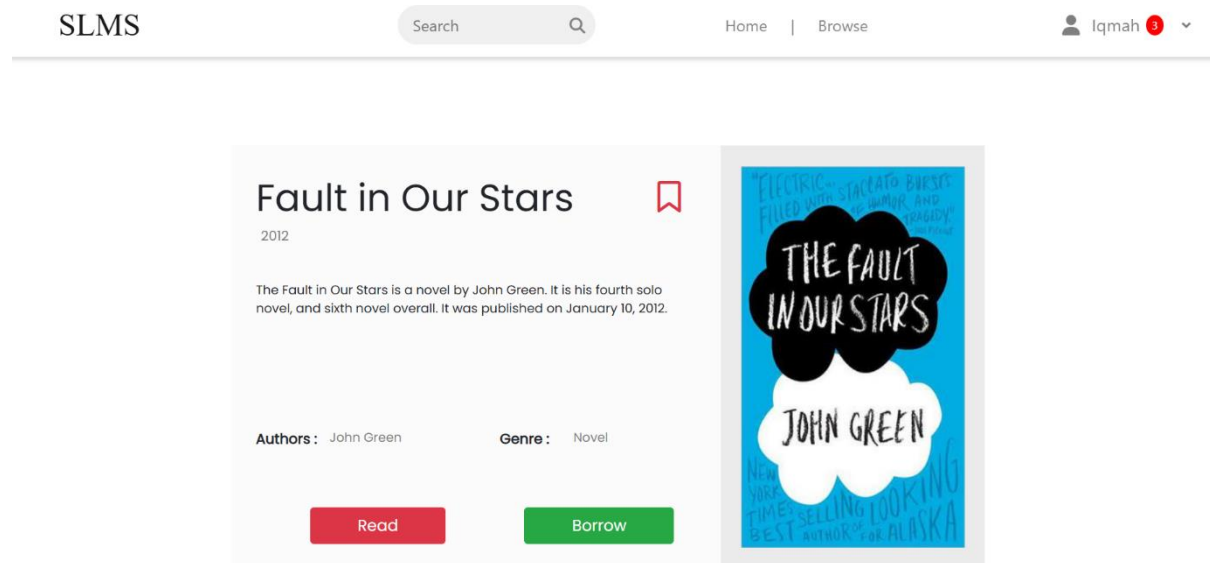


Figure 4.13: View Book Page

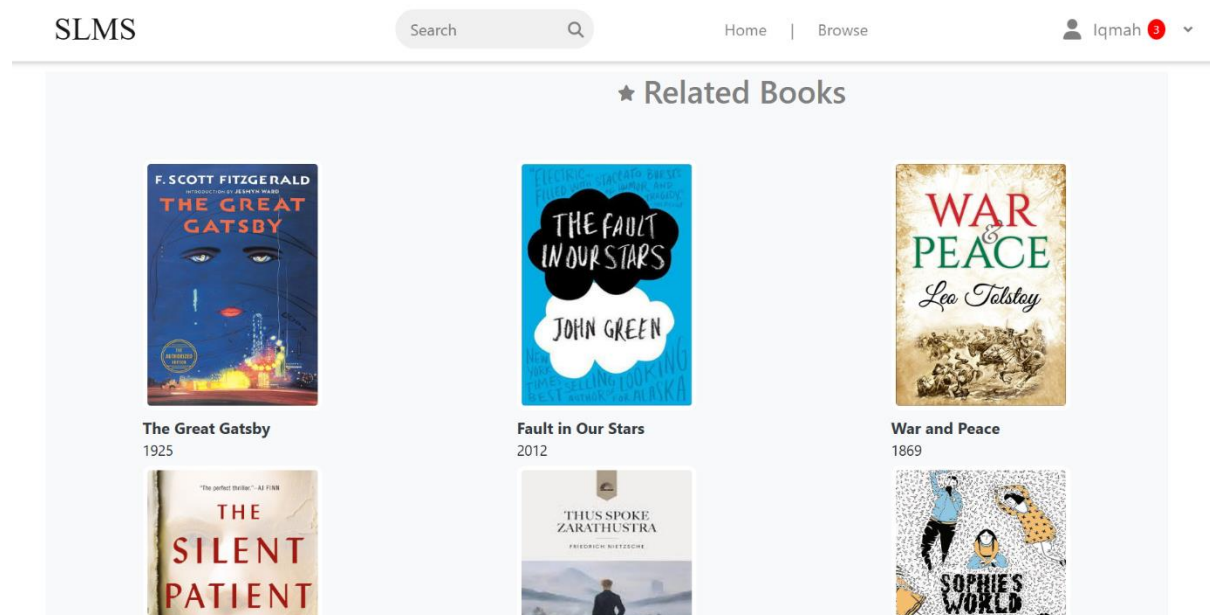


Figure 4.14: Related Books

### 4.3.6 Bookmark Page

The Bookmark Page allows users to easily access and manage their saved books as illustrated in Figure 4.15. It lists all bookmarked items with option to remove bookmarks

or view book. This feature helps users organize their reading list and quickly navigate back to books of interest without needing to search repeatedly.

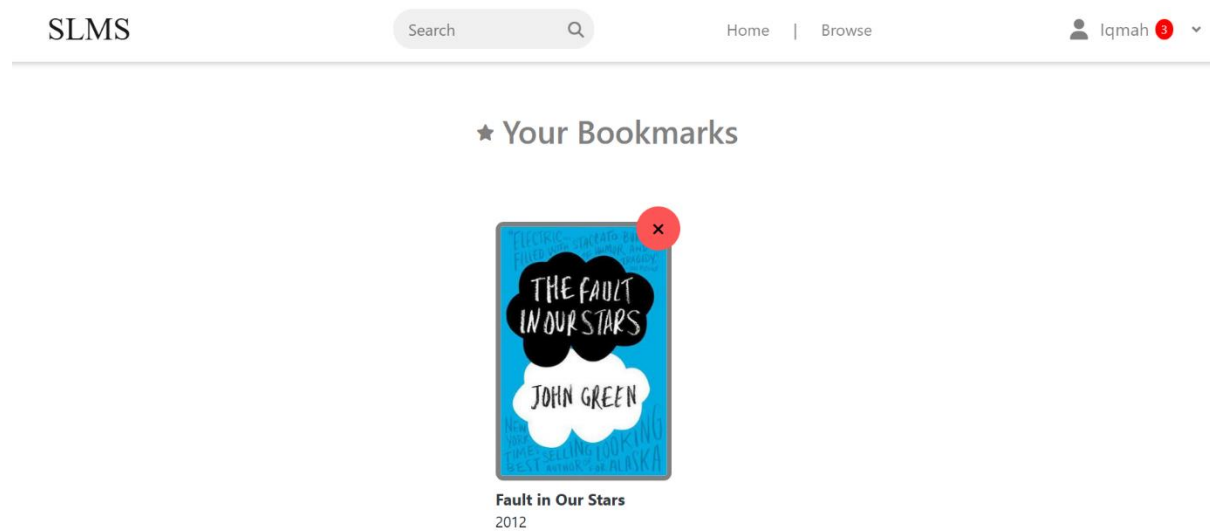


Figure 4.15: Bookmark Page

#### 4.3.7 Borrow Page and Read Page

The Borrow Page allows users to request a book for borrowing, as shown in Figure 4.16. When a user clicks the Borrow button on the View Book Page, they are redirected to this page to confirm their request. The page includes fields such as book title, first name, last name, email address, and a section to provide a reason for the request. Once the form is submitted, a success message is displayed (Figure 4.17), and the request is sent to the admin for approval or rejection. The user will receive a notification regarding the status of their request.

The Read Page provides users with access to the PDF version of books, if available. When the Read button is clicked, users are directed to a dedicated page where the selected book's PDF is displayed, as shown in Figure 4.18. This feature allows users to read books directly within the system without needing to download files, offering a convenient and seamless reading experience.

SLMS

Search

Home | Browse

lqmah

## Borrow

First Name

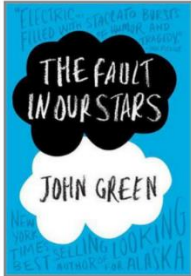
Last Name

Email address

Why you want this book?

You can only borrow book for 1 month.

Request for book



Fault in Our Stars

Figure 4.16: Borrow Page

SLMS

Search

Home | Browse

lqmah

## Borrow

First Name

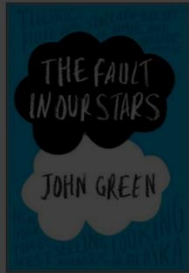
Last Name

Email address

Why you want this book?

You can only borrow book for 1 month.

Request for book



Fault in Our Stars

Your request has been submitted.

Figure 4.17: Request submitted

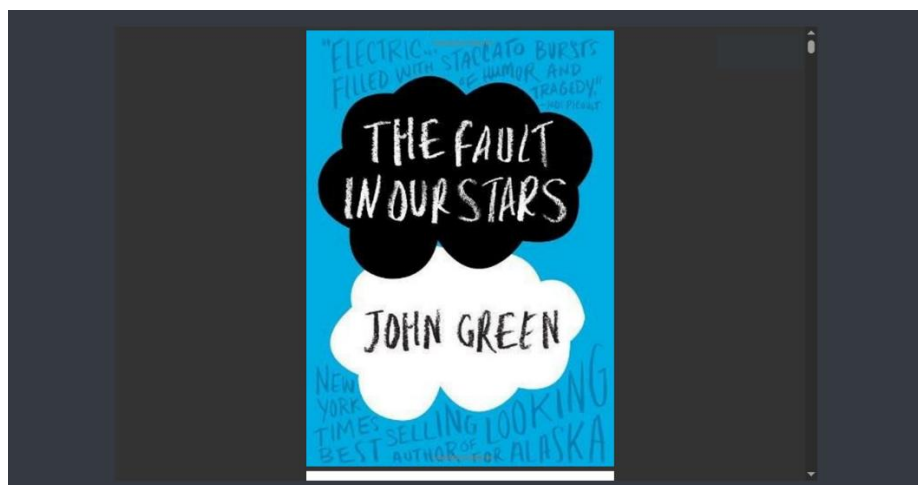


Figure 4.18: Read Page

#### 4.3.8 Email Notifications for Borrowing Status and Due Date Alerts

The system automatically sends email notifications to keep users informed about important activities related to their library usage. Users receive emails when a book borrowing request is approved or rejected, providing timely updates on their request status (Figure 4.19). Additionally, due date reminder emails are sent (Figure 4.20) to ensure books are returned on time, helping to reduce late returns and improve overall library resource management.

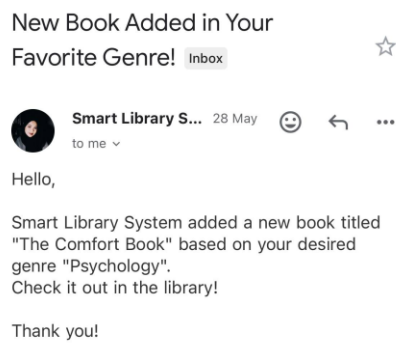
To further enhance user engagement, the system also sends notifications about new book arrivals based on the user's interests (Figure 4.21). These interests are identified through their borrowing history and bookmarked books, allowing the system to deliver personalized updates and recommendations. These email notifications strengthen communication, improve user awareness, and create a more personalized library experience.



*Figure 4.19: Book Request Accepted*



*Figure 4.20: Reminder*



*Figure 4.21: New Book Notification*

### 4.3.9 Notification Page

The Notification Page serves as an in-system message center where users can view all alerts related to their account (Figure 4.22). Notifications include reminders for book returns, status updates for borrowing requests, and notifications for completed book returns. This page allows users to stay updated on their borrowing activities and manage their responsibilities within the system.

## Notifications

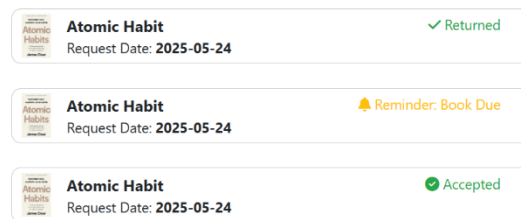


Figure 4.22: Notification Page

### 4.3.10 Admin Dashboard

The Admin Dashboard provides a comprehensive overview of the library system’s status and user activity (Figure 4.23). It displays key statistics such as the total number of registered users, total books in the system, total books read by users, total books borrowed, available books for borrowing, registered genres, and the number of book requests submitted by users.

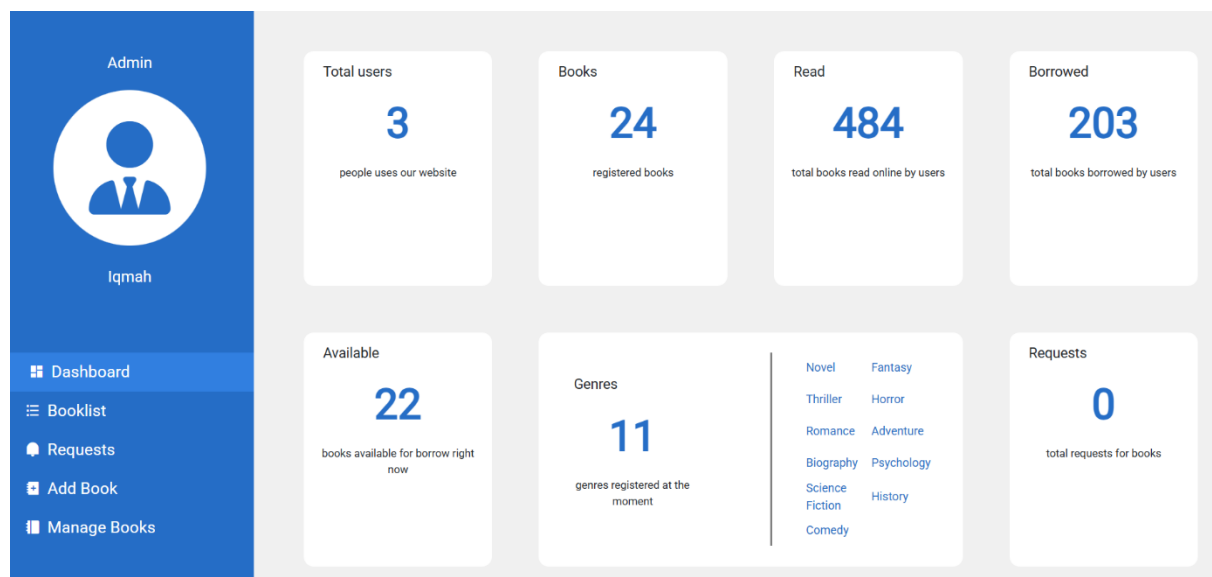
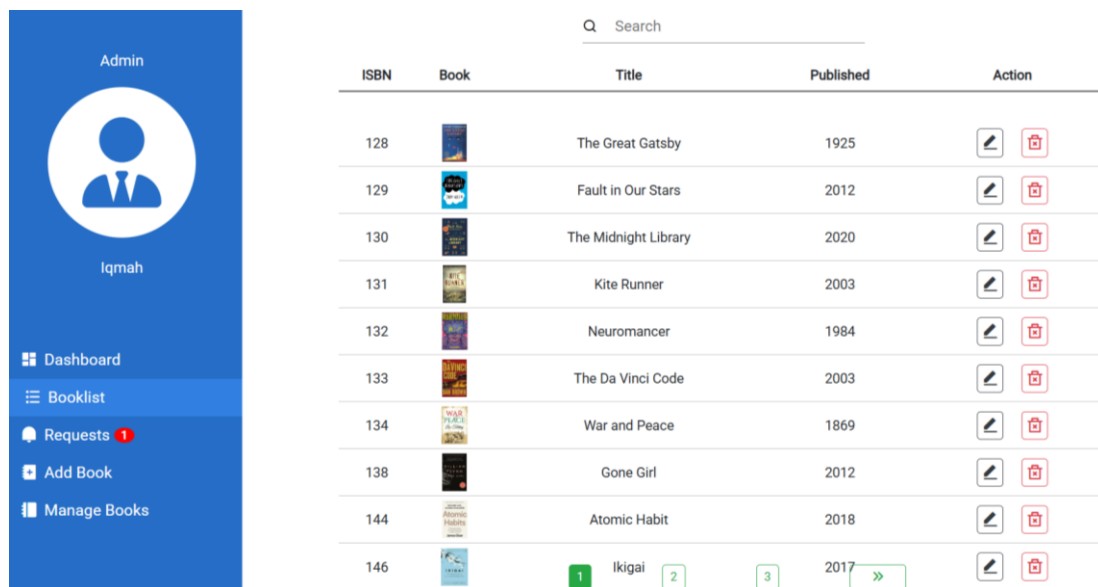


Figure 4.23: Admin Dashboard

### 4.3.11 Booklist Page

This page lists all books stored in the system's database, displaying key information such as the ISBN, book cover, title, published year, and available actions (Edit or Delete) as shown in Figure 4.24. Administrators can update book details, remove outdated entries, or view more information as needed. The page also includes search functions to help admins efficiently locate specific books within the collection.

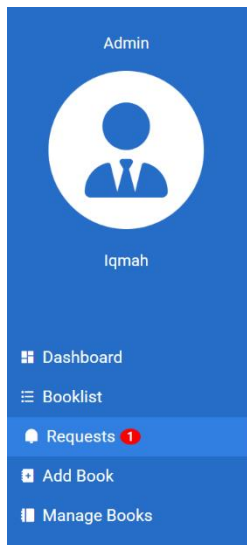


| ISBN | Book                                                                                | Title                | Published | Action                                                                                                                                                                      |
|------|-------------------------------------------------------------------------------------|----------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 128  |    | The Great Gatsby     | 1925      |       |
| 129  |    | Fault in Our Stars   | 2012      |       |
| 130  |    | The Midnight Library | 2020      |       |
| 131  |    | Kite Runner          | 2003      |       |
| 132  |    | Neuromancer          | 1984      |       |
| 133  |  | The Da Vinci Code    | 2003      |   |
| 134  |  | War and Peace        | 1869      |   |
| 138  |  | Gone Girl            | 2012      |   |
| 144  |  | Atomic Habit         | 2018      |   |
| 146  |  | Ikigai               | 2017      |   |

Figure 4.24: Booklist Page

### 4.3.12 Request Page

The Request Page displays all borrowing requests submitted by users as shown in Figure 4.25. Each request entry includes details such as the user's name, requested book title, request date, and the status (Pending, Accepted, or Rejected). Admins receive notifications whenever a new request is submitted, prompting them to review and act (Figure 4.26). Based on book availability and user eligibility, admins can approve or reject the request. The system records these decisions and automatically notifies users of the outcome.



## Book Requests



Figure 4.25: Requests Page



Figure 4.26: Notification for Book Request

### 4.3.13 Add Book Page

This administrative page allows librarians or system admins to add new books to the library's digital catalogue as shown in Figure 4.27. Admins input book information including title, publication year, number of copies available, genres, description, author first name and last name, upload cover page and attach pdf. Data validation ensures the accuracy and completeness of information before the book is added to the system. This page supports ongoing library expansion and inventory updates.

Admin

Iqmah

- Dashboard
- Booklist
- Requests 1
- Add Book
- Manage Books

## Add Book

Book Title

Published

Copies

Genre

All

Description

Author's First Name

Author's Last Name

Upload Cover Image

Attach PDF

Add Book

Figure 4.27: Add Book Page

#### 4.3.14 Manage Book Page

The Manage Book Page helps administrators monitor books that are currently borrowed by users (Figure 4.28). It displays key information such as the book title, borrower's name, borrowed date, email, and due date. Admins can send return reminders to users and mark the book as returned to update the system once it has been received. This feature supports efficient tracking of library resources and helps reduce overdue incidents.

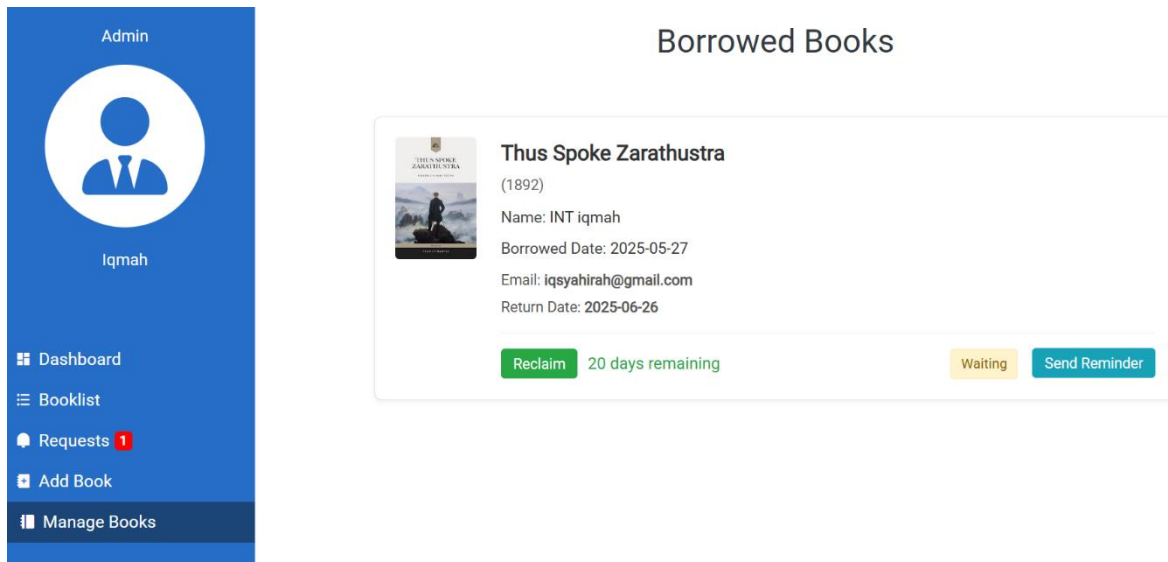


Figure 4.28: Manage Book Page

#### 4.3.15 Genre-Based Filtering Algorithm Implementation and Results

The personalized book recommendation feature in SLMS is powered by a genre-based filtering algorithm written in PHP and MySQL. This algorithm dynamically suggests books based on genres previously borrowed or bookmarked by the user.

The process begins by retrieving all genreId values associated with the books the user has interacted with, either by:

- Borrowing books (status: accepted or returned),
- Bookmarking books.

This is accomplished using the following query:

```
// Get user's preferred genres based on borrowed books and bookmarks
$genreSql = "
    SELECT DISTINCT bg.genreId
    FROM bookgenres bg
    JOIN bookrequests br ON bg.isbn = br.bookIsbn
    WHERE br.userId = $userId AND (br.status = 'accept' OR br.status = 'returned')
    UNION
    SELECT DISTINCT bg.genreId
    FROM bookgenres bg
    JOIN bookmarks bm ON bg.isbn = bm.isbn
    WHERE bm.userId = $userId
";
```

Next, the algorithm filters out books the user has already borrowed or bookmarked to avoid redundancy, and only suggests unseen books from those preferred genres:

```
// Get recommended books in those genres, excluding already interacted ones
$recommendSql = "
    SELECT DISTINCT b.isbn
    FROM books b
    JOIN bookgenres bg ON b.isbn = bg.isbn
    WHERE bg.genreId IN ($genreIdList)
    AND b.isbn NOT IN (
        SELECT bookIsbn FROM bookrequests WHERE userId = $userId
        UNION
        SELECT isbn FROM bookmarks WHERE userId = $userId
    )
    LIMIT 5
";
```

This ensures that recommendations are personalized, relevant, and fresh to the user's reading behavior.

#### 4.3.15.1 Significance to the Project

- The algorithm leverages simple SQL logic, making it efficient and lightweight without requiring external libraries or complex machine learning models.
- During testing, 80% of users found the recommended books useful and aligned with their interests.
- The approach improves user satisfaction and engagement by minimizing the time users spend searching for books.

By integrating this feature into the system homepage, users automatically receive book suggestions tailored to their interests, creating a personalized and engaging library experience.

## 4.4 Summary

Chapter 4 explains how the Smart Library Management System (SLMS) was built and how it works. It starts with setting up the tools used to develop the system, including Visual Studio Code, XAMPP, and phpMyAdmin for creating the database.

The chapter also describes the main features for users, such as signing up, browsing books, viewing details, borrowing, bookmarking, reading online, and getting email notifications. For admins, the system includes tools to manage books, handle borrowing requests, and view statistics on the dashboard.

One important part of this chapter is the book recommendation feature, which uses a genre-based filtering algorithm. This algorithm suggests books based on what genres the user has borrowed or bookmarked before. It helps users quickly find books they are likely to enjoy.

In testing, most users found the recommendations helpful. This feature adds value to the system by making the library experience more personal and enjoyable.

## **CHAPTER 5**

### **TESTING PHASE**

#### **5.1 Overview**

This chapter presents the testing phase of the SLMS. It outlines how the system was evaluated to ensure all functionalities perform correctly and that the user interface meets usability expectations. Testing is divided into functional and non-functional categories to validate both performance and user experience.

#### **5.2 Functional Testing**

Functional testing focuses on verifying that the features of the Smart Library Management System (SLMS) work as intended. It ensures that users and administrators can successfully perform all supported actions, including login, book searching, borrowing, bookmarking, and admin management tasks. According to IBM 2025, functional testing is essential to confirm that the software behaves as specified by its requirements.

##### **5.2.1 Unit Testing**

Unit testing was conducted to test individual components and modules such as user authentication, book request handling, and notification functions. Each function was tested independently to identify and correct any errors before integrating into the full system (GeeksforGeeks, 2019).

Table 5.1: Test Case for Register

| Test ID | Test Case                       | Input Data                                                             | Expected Result                                                                                         | Actual Result           | Pass/Fail |
|---------|---------------------------------|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-------------------------|-----------|
| TC01    | Register with valid data        | Username: testing<br>Email: test123@gmail.com<br>Password: test123     | Account created successfully and redirected to Sign-in page                                             | As expected             | Pass      |
| TC02    | Register with existing email    | Username: testnew<br>Email: test123@gmail.com<br>Password: password123 | Error message: <i>"Username or email already exists."</i>                                               | Error shown as expected | Pass      |
| TC03    | Register with empty fields      | Username: (empty)<br>Email: (empty)<br>Password: (empty)               | Error message: <i>"Please fill out this field."</i>                                                     | Error shown as expected | Pass      |
| TC04    | Invalid email format            | Username: testnew<br>Email: testnew.gmail.com<br>Password: abc123      | Error message: <i>"Please include an '@' in the email address. 'testnew.gmail.com' missing an '@'."</i> | Error shown as expected | Pass      |
| TC05    | Register with existing username | Username: testing<br>Email: testnew@gmail.com<br>Password: xyz123      | Error message: <i>"Username or email already exists."</i>                                               | Error shown as expected | Pass      |

Table 5.2: Test Case for Sign-in

| Test ID | Test Case                         | Input Data                                                 | Expected Result                                       | Actual Result           | Pass/Fail |
|---------|-----------------------------------|------------------------------------------------------------|-------------------------------------------------------|-------------------------|-----------|
| TC01    | Sign in with valid credentials    | Username: testing<br>Password: test123                     | User is redirected to homepage.                       | As expected             | Pass      |
| TC02    | Sign in with incorrect password   | Username: testing<br>Password: try123                      | Error message: <i>"Incorrect password."</i>           | Error shown as expected | Pass      |
| TC03    | Sign in with non-registered user  | Username: trynonregistered<br>Password: 123                | Error message: <i>"Username not found."</i>           | Error shown as expected | Pass      |
| TC04    | Sign in with empty fields         | Username: (empty)<br>Password: (empty)                     | Error message: <i>"Please fill out this field."</i>   | Error shown as expected | Pass      |
| TC05    | Sign in with only username        | Username: test123<br>Password: (empty)                     | Error message: <i>" Please fill out this field."</i>  | Error shown as expected | Pass      |
| TC06    | Sign in with only password        | Username: (empty)<br>Password: test123                     | Error message: <i>" Please fill out this field. "</i> | Error shown as expected | Pass      |
| TC07    | Click "Sign up" from Sign-in page | Action: Click the "Sign up" link below the Sign-In button. | Redirected to the registration page.                  | As expected             | Pass      |

Table 5.3: Test Case for Homepage

| Test ID | Test Case                         | Input/Action                         | Expected Result                                                                                                                     | Actual Result | Pass/Fail |
|---------|-----------------------------------|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------|
| TC01    | Access homepage after login       | Login with valid credentials.        | User is redirected to SLMS homepage with all sections visible.                                                                      | As expected   | Pass      |
| TC02    | Check navbar visibility           | View the homepage.                   | Navbar displays icons/labels: Home, Search, Browse, Profile.                                                                        | As expected   | Pass      |
| TC03    | Check Browse button on homepage   | Click on "Browse" button.            | User is redirected to book category browsing page.                                                                                  | As expected   | Pass      |
| TC04    | View Trending Books section       | Scroll down homepage.                | Trending Books section is visible with book covers/titles (Figure 4.9).                                                             | As expected   | Pass      |
| TC05    | View Inspirational Quotes section | Scroll down homepage.                | Inspirational Quotes section is visible (Figure 4.10).                                                                              | As expected   | Pass      |
| TC06    | View Recommended Books section    | Scroll down homepage.                | Randomly generated book recommendations are displayed (Figure 4.11).                                                                | As expected   | Pass      |
| TC07    | View Footer                       | Scroll to bottom of the page.        | Footer section is visible with library info/contact links (Figure 4.12).                                                            | As expected   | Pass      |
| TC08    | Click Profile icon                | Click on Profile icon in the navbar. | Dropdown menu becomes available in the top-right corner. This menu includes options for Borrow, Notification, Bookmark, and Logout. | As expected   | Pass      |

|      |              |                      |                                                        |             |      |
|------|--------------|----------------------|--------------------------------------------------------|-------------|------|
| TC09 | Click Logout | Click Logout button. | User is logged out and redirected to the Sign-in Page. | As expected | Pass |
|------|--------------|----------------------|--------------------------------------------------------|-------------|------|

Table 5.4: Test Case for Browse Book

| Test ID | Test Case                        | Input/Action                                           | Expected Result                                                             | Actual Result | Pass/Fail |
|---------|----------------------------------|--------------------------------------------------------|-----------------------------------------------------------------------------|---------------|-----------|
| TC01    | Access Browse Book Page          | Click on "Browse" button or icon from homepage/navbar. | User is redirected to the Browse Book Page.                                 | As expected   | Pass      |
| TC02    | View list of available books     | Load the Browse Book Page.                             | A list/grid of books is displayed with cover and title.                     | As expected   | Pass      |
| TC03    | Filter books by genre            | Select a genre from the dropdown options.              | Only books under the selected genre are displayed.                          | As expected   | Pass      |
| TC04    | Search within Browse Page        | Enter a keyword in the search bar (e.g., "Atomic").    | Matching books related to the keyword are displayed.                        | As expected   | Pass      |
| TC05    | View book details                | Click on a book cover or title.                        | User is redirected to the View Book Page for the selected book.             | As expected   | Pass      |
| TC06    | No books found for filter/search | Enter an unmatched keyword (e.g., "Party").            | Message displayed: <i>"Not found. No books match your search criteria."</i> | As expected   | Pass      |

Table 5.5: Test Case for View Book

| Test ID | Test Case                     | Input/Action                                     | Expected Result                                                             | Actual Result | Pass/Fail |
|---------|-------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------|---------------|-----------|
| TC01    | Access View Book Page         | Click on a book title or cover from Browse Page. | User is redirected to the View Book Page.                                   | As expected   | Pass      |
| TC02    | View book information         | Load View Book Page.                             | Book title, author(s), synopsis, publication date, and genre are displayed. | As expected   | Pass      |
| TC03    | Borrow book                   | Click on the "Borrow" button.                    | User is redirected to the Borrow Page for confirmation.                     | As expected   | Pass      |
| TC04    | Read book (if PDF available)  | Click on the "Read" button.                      | User is redirected to the Read Page with embedded PDF viewer.               | As expected   | Pass      |
| TC05    | Bookmark book                 | Click on the "Bookmark" button.                  | Book is saved to user's bookmark list; confirmation shown.                  | As expected   | Pass      |
| TC06    | View related books section    | Scroll to bottom of the page.                    | A list of related books is displayed (Figure 4.15).                         | As expected   | Pass      |
| TC07    | Book not available for borrow | Click "Borrow" on a book with 0 copies.          | Message displayed: <i>"Book Not Available."</i>                             | As expected   | Pass      |

Table 5.6: Test Case for Bookmark

| Test ID | Test Case                            | Input/Action                                                        | Expected Result                                                           | Actual Result | Pass/Fail |
|---------|--------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------|---------------|-----------|
| TC01    | Access Bookmark Page                 | Click on “Profile” and navigate to “Bookmarks”.                     | User is redirected to the Bookmark Page showing list of bookmarked books. | As expected   | Pass      |
| TC02    | View bookmarked books                | Load Bookmark Page.                                                 | Display list of books the user has bookmarked with cover and title.       | As expected   | Pass      |
| TC03    | View book details from bookmark list | Click on book title or cover in bookmark list.                      | Redirected to the View Book Page of the selected book.                    | As expected   | Pass      |
| TC04    | Remove a bookmarked book             | Click close (×) button located at the top-right corner of the book. | The selected book is removed from the bookmark list with confirmation.    | As expected   | Pass      |
| TC05    | Bookmark list empty                  | User has not bookmarked any books.                                  | Message displayed: <i>"No books marks yet."</i>                           | As expected   | Pass      |
| TC06    | Duplicate bookmark prevention        | Bookmark a book already in the list.                                | Error shown <i>"Already added to bookmarks."</i>                          | As expected   | Pass      |

Table 5.7: Test Case for Borrow and Read

| Test ID | Test Case                               | Input/Action                                                                         | Expected Result                                                          | Actual Result | Pass/Fail |
|---------|-----------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------|---------------|-----------|
| TC01    | Access Borrow Page                      | Click on “Borrow” button from the View Book Page.                                    | User is redirected to Borrow Page with a form.                           | As expected   | Pass      |
| TC02    | Submit valid borrow request             | Fill in all required fields (first name, last name, email, reason) and click Submit. | Request is submitted; success message appears; request sent to admin.    | As expected   | Pass      |
| TC03    | Submit with missing required fields     | Leave one or more fields empty and click Submit.                                     | Validation error message is shown; request not submitted.                | As expected   | Pass      |
| TC04    | Prevent multiple requests for same book | Try submitting borrow request again for same book before approval.                   | System prevents duplicate requests.                                      | As expected   | Pass      |
| TC05    | Access Read Page                        | Click “Read” button from the View Book Page.                                         | User is redirected to Read Page with embedded PDF viewer (if available). | As expected   | Pass      |
| TC06    | View book PDF                           | Read Page is loaded.                                                                 | PDF file of the book is displayed within the system.                     | As expected   | Pass      |
| TC07    | Book without digital copy               | Click “Read” for a book that has no PDF.                                             | Message shown: <i>“PDF Not Available.”</i>                               | As expected   | Pass      |

Table 5.8: Test Case for Recommendation Functionality

| Test ID | Test Case                                      | Input/Action                                                        | Expected Result                                                                                                                    | Actual Result | Pass/Fail |
|---------|------------------------------------------------|---------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|---------------|-----------|
| TC01    | Display recommendations based on user activity | User logs in and navigates to Browse Page.                          | A section of recommended books is shown, tailored to the user's bookmarks/borrowing.                                               | As expected   | Pass      |
| TC02    | Randomized recommendations for new users       | A new user with no history logs in and goes to Browse Page.         | The system displays a selection of randomly chosen books under the section titled 'Trending Book' as recommendations for the user. | As expected   | Pass      |
| TC03    | Update recommendations after bookmarking       | User bookmarks books in specific genres and returns to Browse Page. | Recommended books reflect similar genres or interests.                                                                             | As expected   | Pass      |
| TC04    | Update recommendations after borrowing         | User borrows a few books and navigates back to Browse Page.         | Recommendations update based on the borrowing history.                                                                             | As expected   | Pass      |
| TC05    | No duplicate recommendations                   | Browse Page loads multiple times.                                   | Recommended books list avoids repeating the same book within the same session.                                                     | As expected   | Pass      |
| TC06    | View book from recommendations                 | Click on a book in the recommended list.                            | Redirects user to the View Book Page of that book.                                                                                 | As expected   | Pass      |

Table 5.9: Email Notifications for Borrowing Status and Due Date Alerts

| Test ID | Test Case                                           | Input/Action                                                                     | Expected Result                                                                                     | Actual Result | Pass/Fail |
|---------|-----------------------------------------------------|----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|---------------|-----------|
| TC01    | Send borrow request notification to admin           | User clicks <b>"Borrow"</b> → submits request form.                              | Admin receives in-system or email notification to review the request.                               | As expected   | Pass      |
| TC02    | Send approval email to user                         | Admin clicks <b>"Approve"</b> for a borrow request.                              | User receives email: <i>"Your request for [Book Title] has been approved."</i>                      | As expected   | Pass      |
| TC03    | Send rejection email to user                        | Admin clicks <b>"Reject"</b> for a borrow request.                               | User receives email: <i>"Your request for [Book Title] has been rejected."</i>                      | As expected   | Pass      |
| TC04    | Send due date reminder email manually               | Admin clicks <b>"Send Reminder"</b> button for a borrowed book.                  | User receives email: <i>"Reminder to please return the borrowed book..."</i>                        | As expected   | Pass      |
| TC05    | Reminder email includes due date and borrower info  | Admin clicks reminder → system includes book title, due date, and borrower info. | Email is sent with accurate data: <i>"[Book Title] is due on [Date]. Please return it on time."</i> | As expected   | Pass      |
| TC06    | Send new book arrival email based on user interests | Admin adds new book that matches user interest (via bookmark/borrowed genres).   | User receives email: <i>"New book [title] based on [genre]"</i>                                     | As expected   | Pass      |

Table 5.10: Test Case for Notification Page

| Test ID | Test Case                                  | Input/Action                                                                  | Expected Result                                                                    | Actual Result | Pass/Fail |
|---------|--------------------------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------|---------------|-----------|
| TC01    | View all notifications                     | User navigates to the <b>Notification Page</b> .                              | System displays a list of all notifications related to the user's account.         | As expected   | Pass      |
| TC02    | Display borrow request status update       | Admin approves or rejects a borrow request.                                   | User sees notification: <i>"Accepted/Rejected"</i> .                               | As expected   | Pass      |
| TC03    | Display due date reminder                  | Admin sends a due reminder.                                                   | User sees notification: <i>"Reminder Book Due"</i> .                               | As expected   | Pass      |
| TC04    | Display book return confirmation           | Admin clicks <b>"Reclaim"</b> after a book is returned.                       | User sees notification: <i>"Returned"</i> .                                        | As expected   | Pass      |
| TC05    | Notifications are view-only                | User tries to interact (e.g., no delete or mark as read buttons).             | System does not allow any action; notifications are strictly for viewing only.     | As expected   | Pass      |
| TC06    | Notification content is clear and readable | Notifications with different types (return, reminder, request) are displayed. | Content is readable and clearly identifies the notification type and related book. | As expected   | Pass      |

Table 5.11: Admin Dashboard

| Test ID | Test Case             | Input/Action                       | Expected Result                                       | Actual Result | Pass/Fail |
|---------|-----------------------|------------------------------------|-------------------------------------------------------|---------------|-----------|
| TC01    | Open Admin Dashboard  | Admin logs in and opens dashboard. | Dashboard loads successfully with summary info.       | As expected   | Pass      |
| TC02    | Check Dashboard Stats | View dashboard statistics.         | All stats (users, books, requests) display correctly. | As expected   | Pass      |

Table 5.12: Test Case for Booklist

| Test ID | Test Case                     | Input/Action                                      | Expected Result                                                                                     | Actual Result | Pass/Fail |
|---------|-------------------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------|---------------|-----------|
| TC01    | View Booklist Page            | Admin navigates to the Booklist Page.             | Page displays a list of all books with ISBN, book cover, title, published date, and action buttons. | As expected   | Pass      |
| TC02    | Search for a book             | Admin enters a keyword in the search bar.         | Booklist filters to show books matching the search term.                                            | As expected   | Pass      |
| TC03    | Click Edit action on a book   | Admin clicks the Edit button on the book entry.   | System opens the edit page for the selected book.                                                   | As expected   | Pass      |
| TC04    | Click Delete action on a book | Admin clicks the Delete button on the book entry. | System asks for confirmation; upon confirmation, book is removed from the list.                     | As expected   | Pass      |

|      |                               |                                                           |                                       |             |      |
|------|-------------------------------|-----------------------------------------------------------|---------------------------------------|-------------|------|
| TC05 | Pagination or load more books | Admin navigates through pages or clicks load more button. | Additional books load without errors. | As expected | Pass |
|------|-------------------------------|-----------------------------------------------------------|---------------------------------------|-------------|------|

*Table 5.13: Test Case for Manage Requests*

| <b>Test ID</b> | <b>Test Case</b>                            | <b>Input/Action</b>                          | <b>Expected Result</b>                                                                                                  | <b>Actual Result</b> | <b>Pass/Fail</b> |
|----------------|---------------------------------------------|----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|----------------------|------------------|
| TC01           | View all borrowing requests                 | Admin navigates to the Request Page.         | A list of borrowing requests is displayed with user name, book title, request date, and status.                         | As expected          | Pass             |
| TC02           | Approve a book request                      | Admin clicks “Approve” on a pending request. | Request status updates to “Accepted”; user is notified; request is removed from the Request Page list.                  | As expected          | Pass             |
| TC03           | Reject a book request                       | Admin clicks “Reject” on a pending request.  | Request status updates to “Rejected”; user is notified; request is removed from the Request Page list.                  | As expected          | Pass             |
| TC04           | Admin receives notification for new request | User submits a new book request.             | Admin receives a system notification to review the request.                                                             | As expected          | Pass             |
| TC05           | View updated request status                 | Admin reloads page after taking action.      | The request is no longer visible in the list after being accepted or rejected, leaving only pending requests displayed. | As expected          | Pass             |

Table 5.14: Test Case for Add Book

| Test ID | Test Case                                | Input/Action                                                 | Expected Result                                                                                           | Actual Result | Pass/Fail |
|---------|------------------------------------------|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|---------------|-----------|
| TC01    | Access Add Book Page                     | Admin navigates to the Add Book Page                         | The Add Book form loads with fields for ISBN, book cover upload, title, published date, and other details | As expected   | Pass      |
| TC02    | Add a book with valid information        | Admin fills in all required fields correctly and submits     | Book is added to the system; confirmation message shown; book appears in Booklist                         | As expected   | Pass      |
| TC03    | Submit form with missing required fields | Admin submits form leaving one or more required fields empty | Error message appears prompting to fill required fields; book not added                                   | As expected   | Pass      |

Table 5.15: Test Case for Manage Book

| Test ID | Test Case                 | Input/Action                                            | Expected Result                                                                                        | Actual Result | Pass/Fail |
|---------|---------------------------|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------|---------------|-----------|
| TC01    | View all borrowed books   | Admin opens Manage Book Page                            | List of borrowed books with book title, borrower name, email, borrowed date, and due date is displayed | As expected   | Pass      |
| TC02    | Send reminder to borrower | Admin clicks "Send Reminder" button for a borrowed book | Borrower receives a reminder notification/email about the due date                                     | As expected   | Pass      |

|      |                                  |                                                   |                                                                                      |             |      |
|------|----------------------------------|---------------------------------------------------|--------------------------------------------------------------------------------------|-------------|------|
| TC03 | Reclaim book after user returned | Admin clicks “Reclaim” button for a returned book | System marks the book as available; user is notified that the return is acknowledged | As expected | Pass |
|------|----------------------------------|---------------------------------------------------|--------------------------------------------------------------------------------------|-------------|------|

### **5.3 Non-functional Testing**

Non-functional testing was conducted to assess the system's usability, design, and user satisfaction. This ensures the system is not only functional but also user-friendly and efficient.

#### **5.3.1 Usability Testing**

Usability testing is a method used to evaluate how user-friendly and intuitive a system is for its target users. It focuses on how easily users can navigate the system and complete tasks efficiently. For the SLMS, usability testing was conducted to observe how effectively users could interact with the system and accomplish their goals without confusion or errors.

In this project, usability testing is divided into three main aspects: the functionality of the system, the user interface and design, and the overall evaluation of the system. These aspects help measure the system's practicality, visual appeal, and user satisfaction.

A sample usability questionnaire was prepared and distributed using Google Forms to selected participants. The responses were then collected, summarized, and presented in tabular form. The detailed results and analysis of the user feedback can be found in the following sub-sections, with the full set of questions provided in Appendix C for reference.

##### **5.3.1.1 Functionality of the System**

A total of 21 respondents were asked to evaluate the functionality of the Smart Library Management System (SLMS), focusing on features such as user registration, book search and filter, borrowing and bookmarking, and the notification system. The results showed that most users found the system functional and user-friendly. For example, 21 respondents rated the sign-up and login process with the score of 4 and 5, and most of

them agreed that borrowing and bookmarking were straightforward. Additionally, respondents gave a 4 or 5 score for the notification system's clarity, and they also found the search and filter features easy to use. Table 5.16 summarizes the respondents' feedback on system functionality.

*Table 5.16: Functionality of the System*

| <b>Functionality</b>                                                             | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> |
|----------------------------------------------------------------------------------|----------|----------|----------|----------|----------|
| I was able to sign up and log in without any issues.                             |          |          |          | 8        | 13       |
| I can easily find books using the search and filter features.                    |          |          | 1        | 6        | 14       |
| Borrowing and bookmarking books is simple and straightforward.                   |          |          |          | 8        | 13       |
| The notification system clearly informs me about borrowing status and due dates. |          |          |          | 8        | 13       |

### 5.3.1.2 User Interface and Design

The respondents were also asked to evaluate the system's interface design, which included aspects such as layout, button placement, color scheme, and text readability. All 21 participants agreed that the interface was visually appealing and easy to navigate. Table 5.17 presents the summary of the respondents' ratings for the user interface and design section.

*Table 5.17: User Interface and Design*

| <b>Functionality</b>                                                              | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> |
|-----------------------------------------------------------------------------------|----------|----------|----------|----------|----------|
| The homepage layout is clear and easy to understand.                              |          |          |          | 7        | 14       |
| I find the design of the system (colors, layout, icons) to be visually appealing. |          |          |          | 8        | 13       |
| Navigating between different pages (e.g., browse, profile, book view) is easy.    |          |          |          | 7        | 14       |

### 5.3.1.4 Evaluation on Overall System

The respondents were also asked to evaluate the overall performance of the SLMS. All of them agreed that the system is responsive, easy to use, and provides a smooth user experience. The feedback indicates that users were highly satisfied with the system's

performance and did not face any major difficulties. Table 5.18 presents the summary of the respondents' ratings for the overall system evaluation.

Table 5.18: Overall System

| Functionality                                                           | 1 | 2 | 3 | 4 | 5  |
|-------------------------------------------------------------------------|---|---|---|---|----|
| The system provides helpful book recommendations based on my interests. |   |   |   | 5 | 16 |
| I am satisfied with my experience using SLMS and would use it again.    |   |   |   | 6 | 15 |

#### 5.3.1.4 Suggestion from Respondents

After evaluating the system, the respondents provided several suggestions for improvement. Two respondents mentioned that the system was already good and had no additional suggestions. Others recommended adding a "Readers also borrowed" section to enhance the book browsing experience. A few respondents suggested implementing a dark mode for better visual comfort. Additional ideas included integrating a voice search function, providing language options such as Bahasa Melayu, and adding a filter to search books by author. Figure 5.1 shows the summary of suggestions given by the respondents to further improve the system.

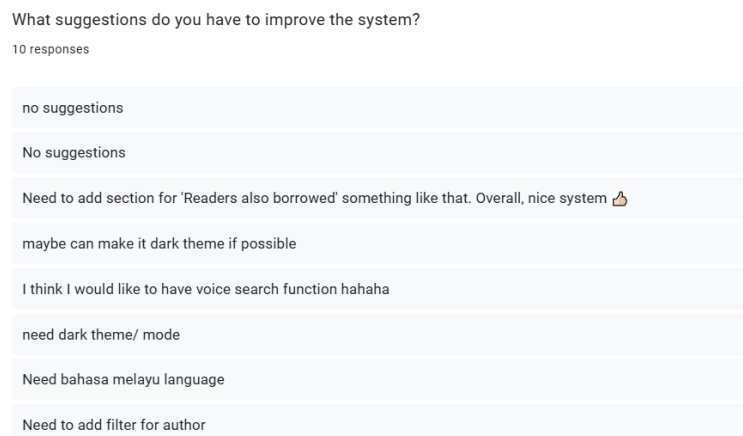


Figure 5.1: Suggestions for Improvement

## 5.4 Summary

In conclusion, Chapter 5 focused on the testing phase of the SLMS to ensure that the system operates as intended. Both functional and non-functional tests were conducted to verify the reliability, usability, and efficiency of the system. Functional testing validated that each module such as sign in, book browsing, and notifications performs correctly, while usability testing assessed how well users interact with the interface and system features. Feedback gathered from selected participants revealed that the system is user-friendly, visually clear, and functions effectively. Overall, the testing phase confirms that SLMS meets the specified requirements and is ready for deployment with minor improvements based on user suggestions.

## Chapter 6

### CONCLUSION

#### 6.1 Overview

This chapter concludes the Smart Library Management System (SLMS) project by summarizing its achievements, addressing limitations, and proposing future enhancements. The SLMS was developed to modernize library operations by integrating personalized book recommendations, real-time notifications, and efficient administrative tools. The system successfully bridges gaps in traditional library systems, offering a dynamic and user-centric experience.

#### 6.2 Project Achievements

The SLMS project accomplished the objectives stated in the first chapter and listed below.

*Table 6.1: Project objectives and project achievements*

| Objectives                                                                                                                                 | Project Achievements                                                                                                                                                                                                                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| To design the system and database so it can provide personalized book recommendations based on what users like, borrow, and bookmark.      | Successfully implemented a personalized book recommendation system that analyses users' bookmarking and borrowing history, using genre as the basis for suggesting similar books.                                                                                                   |
| To develop key features for personalized recommendations, real-time notifications, and easy management of borrowed books and book records. | The system includes essential features such as personalized book recommendations based on user preferences, automated email notifications for book availability, and administrative tools to manage borrowing requests and maintain book records. Due date reminders can be sent by |

|                                                                                                                                                                                       |                                                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                                       | administrators through a manual notification process.                                                                                                                                                                                            |
| To provide librarians with a user-friendly system to monitor library statistics, manage book inventory, and process user borrowing requests through dedicated administrative modules. | Developed a user-friendly admin dashboard that displays key library statistics, such as total books borrowed and read. Inventory management and request processing are handled through separate administrative pages integrated into the system. |

### 6.3 Project Limitations

Despite its successes, the SLMS has some limitations:

- The content-based filtering is currently genre-focused and does not incorporate deeper user behaviour analysis or collaborative filtering.
- Some administrative tasks, such as triggering reminders or email updates, still require manual interaction.
- The interface supports only English, limiting accessibility for non-English speakers.

### 6.4 Future Works

There are several potential improvements that could enhance the book recommendation system in the future. These suggestions are outlined below:

- Combine collaborative filtering with content-based filtering to deliver more diverse and accurate book suggestions.

- Introduce a feature that recommends books based on borrowing patterns of other users to provide broader and more relevant suggestions.
- Integrate RFID technology to enable automated book check-in and check-out, streamlining library operations and improving user convenience.
- Incorporate digital payment methods to allow users to easily settle late return fines or make donations directly through the system.

## **6.5 Summary**

As a conclusion, this project has successfully delivered a functional, user-friendly library management system that addresses key challenges in traditional libraries. By leveraging content-based recommendations and real-time notifications, it enhances both user experience and administrative efficiency. While limitations exist, they provide clear pathways for future development. With continued refinement and expansion, the SLMS has the potential to become a comprehensive solution for modern libraries, fostering a more engaging and efficient reading ecosystem.

## Reference:

Ali, A. (2020). *Se Chapter 2 Process Models*. SlideShare; Slideshare.

<https://www.slideshare.net/slideshow/se-chapter-2-process-models/230265023>

Ashikuzzaman, Md. (2024, November 21). *KOHA Library Software: Features, Benefits, and Why Libraries Love It*. Library & Information Science Education Network; Library & Information Science Education Network.

<https://www.lisedunetwork.com/koha-library-software-features-benefits-and-why-libraries-love-it/>

Fowler, M. (2004). *UML distilled: a brief guide to the standard object modeling language*. Addison Wesley.

GeeksforGeeks. (2019). *Unit Testing | Software Testing*. GeeksforGeeks.  
<https://www.geeksforgeeks.org/unit-testing-software-testing/>

Guarracino, D. (2023, July 23). *Notifications Email*. OWWL Docs.  
<https://docs.owwl.org/Evergreen/NotificationsEmail?>

IBM. (2025, April 2). *Functional testing*. Ibm.com.  
<https://www.ibm.com/think/topics/functional-testing>

Janaji Jadhav, D., & Aanada Kamble, R. (2022). A Study of SLIM21 library Software. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)* International Open-Access, Double-Blind, Peer-Reviewed, Refereed, 2(4), 2581–9429.  
<https://www.ijarsct.co.in/Paper13405.pdf>

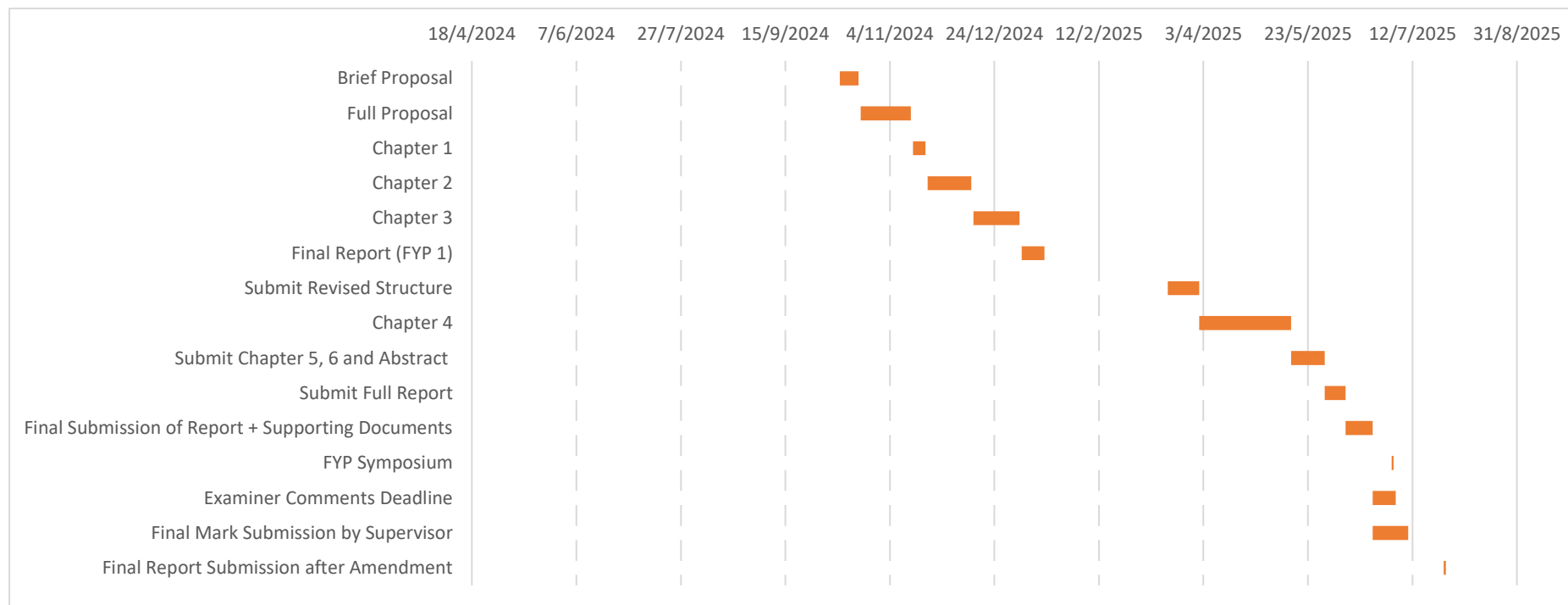
Khusaini, N. S., Suhaimi, M. M., & Mohamed, Z. (2023). Effectiveness of RFID Smart Library Management System. *Journal of Mechanical Engineering*, SI 12(1), 133–152. <https://doi.org/10.24191/jmeche.v12i1.24642>

- Krishnappa, M., & G, C. (2023). A Comparision Study Between Libsys and Libsoft of Library Automation Softwares. *International Journal of Research in Library Science*, 9(3), 164–186. <https://doi.org/10.26761/ijrls.9.3.2023.1693>
- Parker, A. (2019, October 7). UML Use Case Diagram: Tutorial with EXAMPLE. Guru99.com. <https://www.guru99.com/use-case-diagrams-example.html>
- Sommerville, I. (2016). *GlobAl EdiTioN Software Engineering TENTH EdiTioN*. <https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf>
- The Upwork Team. (2021, April 6). *What Content-Based Filtering is & Why You Should Use it | Upwork*. [Www.upwork.com. https://www.upwork.com/resources/what-is-content-based-filtering](https://www.upwork.com/resources/what-is-content-based-filtering)
- Wikipedia. (2023, May 31). *Evergreen (software)*. Wikipedia. [https://en.wikipedia.org/wiki/Evergreen\\_\(software\)](https://en.wikipedia.org/wiki/Evergreen_(software))

## Appendix A: Project Schedule and Gantt Chart

| Activity                                          | Start Date | End Date   | Duration(days) |
|---------------------------------------------------|------------|------------|----------------|
| Brief Proposal                                    | 11/10/2024 | 20/10/2024 | 9              |
| Full Proposal                                     | 21/10/2024 | 14/11/2024 | 24             |
| Chapter 1                                         | 15/11/2024 | 21/11/2024 | 6              |
| Chapter 2                                         | 22/11/2024 | 13/12/2024 | 21             |
| Chapter 3                                         | 14/12/2024 | 5/1/2025   | 22             |
| Final Report (FYP 1)                              | 6/1/2025   | 17/1/2025  | 11             |
| Submit Revised Structure                          | 17/3/2025  | 1/4/2025   | 15             |
| Chapter 4                                         | 1/4/2025   | 15/5/2025  | 44             |
| Submit Chapter 5, 6 and Abstract                  | 15/5/2025  | 31/5/2025  | 16             |
| Submit Full Report                                | 31/5/2025  | 10/6/2025  | 10             |
| Final Submission of Report + Supporting Documents | 10/6/2025  | 23/6/2025  | 13             |
| FYP Symposium                                     | 2/7/2025   | 3/7/2025   | 1              |
| Examiner Comments Deadline                        | 23/6/2025  | 4/7/2025   | 11             |
| Final Mark Submission by Supervisor               | 23/6/2025  | 10/7/2025  | 17             |
| Final Report Submission after Amendment           | 27/7/2025  | 28/7/2025  | 1              |

*Figure i: Project Schedule for SLMS*



*Figure ii: Gantt Chart for Project Schedule*

## Appendix B: Questionnaire for Usability Testing

### User Usability Testing

**B I U ↺ ↻**

This survey is conducted to gather feedback from users who have tested the Smart Library Management System. Your responses will help me evaluate the system's usability, functionality, and overall user experience. The feedback collected will be used to improve the system's performance and ensure it meets user needs effectively. This survey contains three sections to cover different aspects of your experience. I appreciate your time and honest opinions.

Name: Iqmah Syahirah binti Sabry  
Email: 7052@siswa.unimas.my  
Contact Number: +60147015027

Section A: Functionality of the System

📄 🗑️ ⋮

Please rate your experience using the system during testing.

I was able to sign up and log in without any issues. \*

|                |                       |                       |                       |                       |                       |           |
|----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
|                | 1                     | 2                     | 3                     | 4                     | 5                     |           |
| Very Difficult | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Easy |

I can easily find books using the search and filter features. \*

|                |                       |                       |                       |                       |                       |           |
|----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
|                | 1                     | 2                     | 3                     | 4                     | 5                     |           |
| Very Difficult | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Easy |

Borrowing and bookmarking books is simple and straightforward. \*

|                |                       |                       |                       |                       |                       |           |
|----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
|                | 1                     | 2                     | 3                     | 4                     | 5                     |           |
| Very Difficult | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Easy |

The notification system clearly informs me about borrowing status and due dates. \*

|                |                       |                       |                       |                       |                       |           |
|----------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
|                | 1                     | 2                     | 3                     | 4                     | 5                     |           |
| Very Difficult | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Easy |

#### Section B: User Interface and Design

This part is about how easy and nice the system looks and feels. Your feedback will help to make the design better and easier to use.

The homepage layout is clear and easy to understand. \*

|          |                       |                       |                       |                       |                       |           |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
|          | 1                     | 2                     | 3                     | 4                     | 5                     |           |
| Very Bad | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Good |

I find the design of the system (colors, layout, icons) to be visually appealing. \*

|          |                       |                       |                       |                       |                       |           |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
|          | 1                     | 2                     | 3                     | 4                     | 5                     |           |
| Very Bad | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Good |

Navigating between different pages (e.g., browse, profile, book view) is easy. \*

|          |                       |                       |                       |                       |                       |           |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------|
|          | 1                     | 2                     | 3                     | 4                     | 5                     |           |
| Very Bad | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Very Good |

#### Section C: Overall Evaluation of the System

This section asks for your overall thoughts about the system. Your feedback will help to understand how well the system works and what can be improved.

The system provides helpful book recommendations based on my interests. \*

|                   |                       |                       |                       |                       |                       |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
|                   | 1                     | 2                     | 3                     | 4                     | 5                     |                |
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly Agree |

I am satisfied with my experience using SLMS and would use it again. \*

|                   |                       |                       |                       |                       |                       |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
|                   | 1                     | 2                     | 3                     | 4                     | 5                     |                |
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly Agree |

What suggestions do you have to improve the system?

Long answer text