

Library Management System

Muhammad Eshak Azri¹, Rozita Abdul Jalil^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: rozita@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2026.07.01.081>

Article Info

Received: 12 June 2025

Accepted: 18 June 2026

Available online: 30 June 2026

Keywords

Library Management System (LMS),
Web-Based Library System, Library
Automation, Library System
Workflow, Book Borrowing

Abstract

This project focuses on developing a Library Management System for the Parit Haji Siraj Library, which currently uses traditional, manual methods for managing its daily operations. These outdated methods slow down operations and create errors in managing library activities. As for the problem, this project introduces a Library Management System to improve and automate key library processes. The main objectives of this system are to reduce manual work, ensure accurate tracking of resources, and simplify essential tasks like book borrowing and returns. The system provides features like member registration, book catalogues, and overdue notifications, focusing on functionality and reliability. Developed using PHPMyAdmin for database management, along with web technologies, the system ensures secure data handling and operational consistency. The features are simple and practical, making them easy for library staff to use in daily operations. The results show that the system reduces errors in record-keeping, saves time, and increases operational efficiency. Staff can now handle tasks more effectively without depending on manual processes. In conclusion, the Parit Haji Siraj Library Management System provides a simple yet effective solution to improve library management. Future upgrades include digital library access and basic analytics to understand book circulation trends.

1. Introduction

The Parit Haji Siraj Library plays a crucial role in supporting education and literacy in the community by offering a variety of books and resources. As a key public service, the library helps students and residents with their learning needs, providing books and a place for study and engagement.

Libraries serve as educational center, offering books, journals, and digital tools to help users access and apply information. They also host programs, events, and provide quiet study spaces, fostering both academic and personal growth.

In Parit Haji Siraj, the library is a central point for learning, holding events and providing resources to the community. However, the current system for managing the library's operations relies on Microsoft Excel, where staff manually enter data for book inventories, member records, and borrowing transactions. This process is time-consuming and prone to human error, leading to inaccurate records and inefficiencies, especially with overdue books. There are no automated reminders for overdue books, creating confusion for both staff and library users.

This project aims to implement an automated library management system to improve efficiency. The new system will automate tasks such as inventory tracking, member management, and borrowing transactions. Key features will include automatic reminders for overdue books, helping to ensure books are returned on time and improving overall management.

This is an open access article under the CC BY-NC-SA 4.0 license.



2. Related Work

In this section, some existing studies and library management systems (LMS) are discussed. This helps to understand what features are commonly used and what improvements can be made.

2.1 Overview of Library Management Systems

Library Management Systems are essential for libraries nowadays. They help organize books, keep track of members, and manage borrowing and returning activities. Without these systems, library staff would have to do everything manually, which takes time and is more likely to cause mistakes. LMS help by making the process faster and more accurate.

Many LMS include features such as cataloging books, registering users, and tracking book loans. Modern systems often allow users to search for books easily and reserve or hold books online. Web-based systems are becoming popular because they allow access from different devices anytime and anywhere.

2.2 Recent Trends in Library Management Systems

Library systems have improved a lot with new technology. Many libraries now use web-based LMS that can be accessed from phones and tablets, making it easier for users to search and borrow books remotely.

Some systems connect to digital libraries and e-books, giving users more options beyond physical books. Notifications by email or SMS remind users when books are due or overdue, which helps reduce late returns. These improvements help libraries provide better service to users.

2.3 Comparison with the Existing Systems

As part of the development process, several existing library management systems were reviewed to understand their features and how they handle key library functions. This helped identify which features are common and which could be improved or added. Table 1 shows the comparisons between similar systems and the developed system.

Table 1 System Comparison

System Features	Koha	Tunku Tun Aminah Library System	University Malaya's Library System	Developed Library Management System
Login	√	√	√	√
Book Searching/ availability	√	√	√	√
Borrow and return book	√	√	√	√
Book Catalog	√	√	√	√
Notification overdue book	X	√	√	√
Penalty on overdue book	X	√	√	√

3. Methodology

The Agile Model was chosen for the development of the library management system because it is flexible and allows for adjustments based on changing user needs. This iterative approach enables continuous feedback and improvements, ensuring the system remains practical and responsive to the library's evolving requirements. By focusing on features like book management, user profiles, and overdue notifications, Agile allows for minor, incremental updates that align the system with real-world needs, improving the user experience and overall system performance without delaying the project. Table 2 below outlines the software development activities, showing how the Agile methodology is applied at each stage of the system development process.

Table 2 *Software development activities and their task*

Phase	Task	Output
Requirement	Gathered functional and non-functional requirements and identified the system's key features and functionality.	Requirement specifications document and use case diagrams.
Design	Designed the system architecture, database structure, wireframes, and user interface designs.	System design documentation and developed database schema.
Development	Developed the system based on design specifications and integrated components to ensure functionality.	Completed system with working features.
Testing	Conducted unit testing on components, performed integration testing, and tested system usability, performance, and security.	Test reports and fixed identified issues.
Deployment	Configured the production environment, deployed the system, and conducted post-deployment checks and user training.	System deployed in the live environment.
Review	Collected feedback from users, assessed system performance, and identified improvements.	User feedback report and system performance review.

3.1 System Requirements

The system design includes an overview of the functional and non-functional requirements, user expectations. These components are essential for the successful development, implementation, and operation of the system, ensuring it aligns with user needs and meets performance criteria. Tables 3 and 4 present the functional and non-functional requirements, providing further details on how these factors influence the system's design.

Table 3 *Functional Requirements*

No	Module	Description
1	Book Management	Allow admin to add, update, and delete book records.
2	Login and Registration	Allow users to register and update their details. Allow admin to register and update admin details.
3	Borrow and Return Module	Allow admin to process book borrowing and returns. Allow users to view borrowing history and details.
4	Book Searching	Allow users to search for books. Allow users to filter books based on availability.
5	Notification Module	The system sends notifications for overdue books via email.
6	Reporting	The system generates reports on borrowed and overdue books.

Table 4 *Non-Functional Requirements*

No	Requirement	Description
1	Performance	The system should be responsive and handle typical usage smoothly.
2	Operational	The system should load completely and be fully functional under normal operating conditions.
3	Security	User and admin data must be protected against unauthorized access through encryption and secure authentication.
4	Compatibility	The system should be able to work on any web browser.

User requirements define the needs and expectations for the system, focusing on account management, access to functionalities, and interaction with system features. Identifying these requirements is crucial to ensure the system aligns with user expectations and delivers efficient performance. Table 5 provides a detailed explanation of these user requirements.

4. System Analysis and Design

An overview of the findings includes the Data Flow Diagram (DFD) and Entity-Relationship Diagram (ERD), which illustrate the primary processes, data flow, and relationships between system entities. These analyses provide insight into the system's workflow, user interactions, and data management, supporting an efficient design for core library operations such as book borrowing, returns, and user management. Figure 1 presents the context diagram, while Figure 2 provides the DFD Level 0.

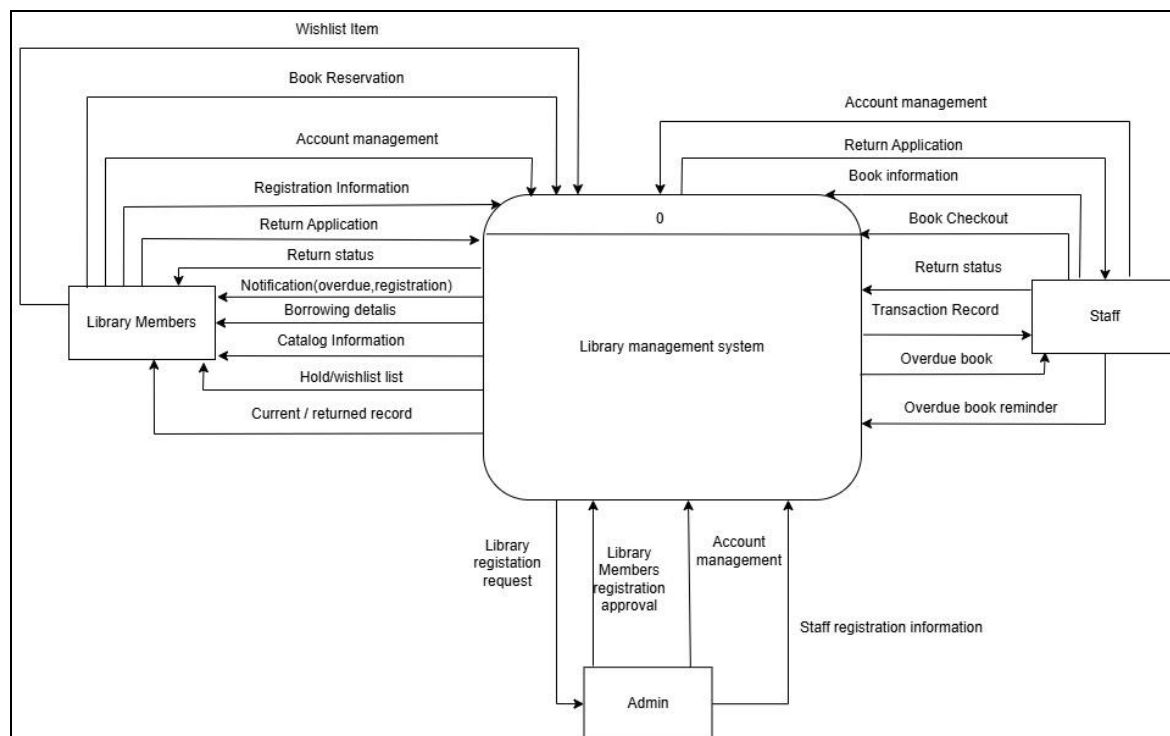


Fig. 1 Context Diagram

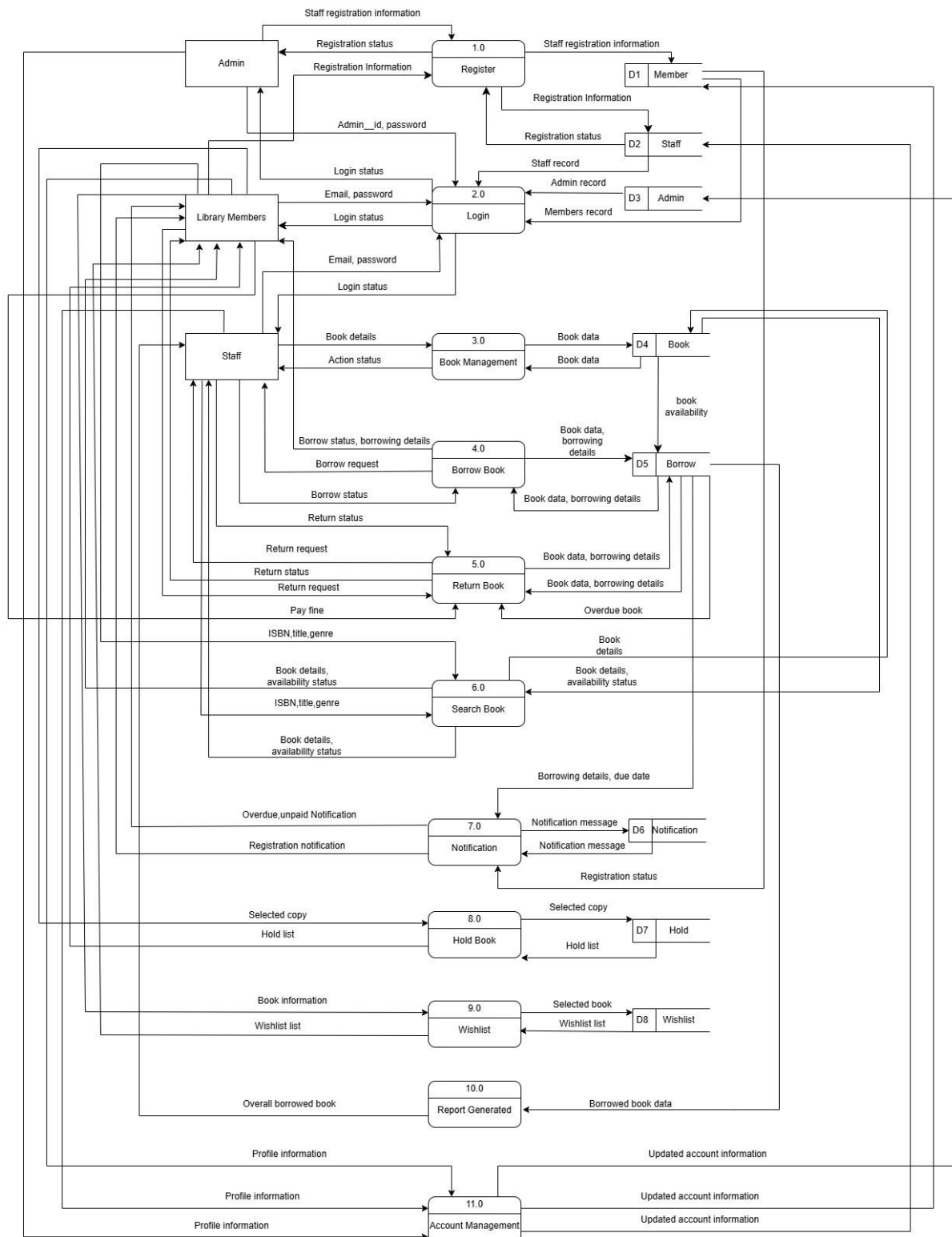


Fig. 2 Data Flow Diagram (DFD)

4.1 Entity-Relationship Diagram (ERD)

An Entity-Relationship Diagram (ERD) illustrates the database structure and the relationships among entities within the system. The system consists of several main tables, including Admin, Staff, Member, Book, Borrow, Book Copies, Wishlist, Hold, and Notification. Key relationships, such as the one-to-many relationship between Member and Borrow, help ensure data consistency and reflect real-world interactions. These connections support the overall functionality of the system. Figure 3 shows the visual representation of these relationships.

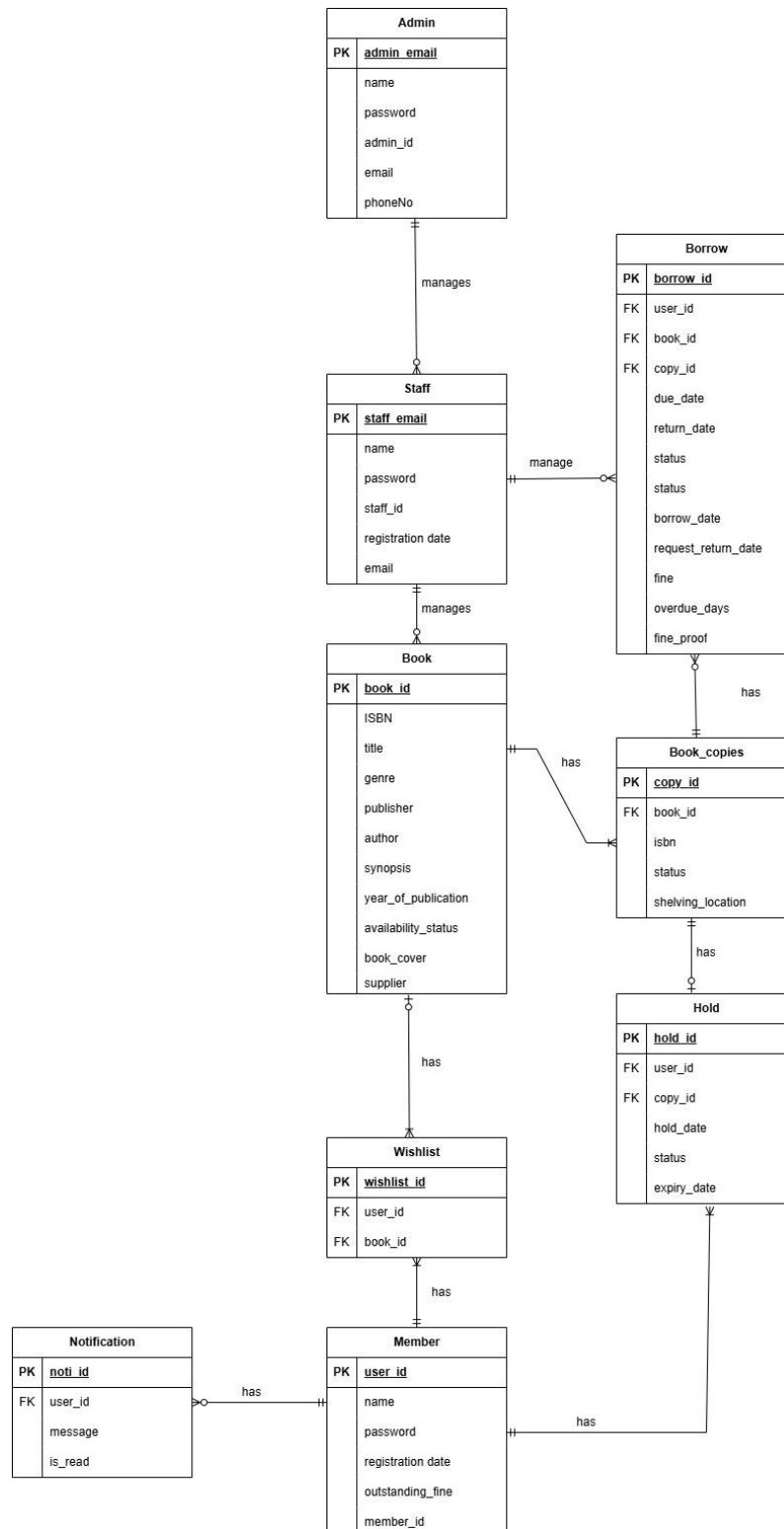


Fig. 3 Entity Relationship Diagram (ERD)

4.2 Flowchart

The flowchart below represents the overall workflow of the Library Management System, outlining the different actions available based on user roles: Library Member, Staff, and Admin. Each role interacts with the system differently, ensuring efficient management and user access.

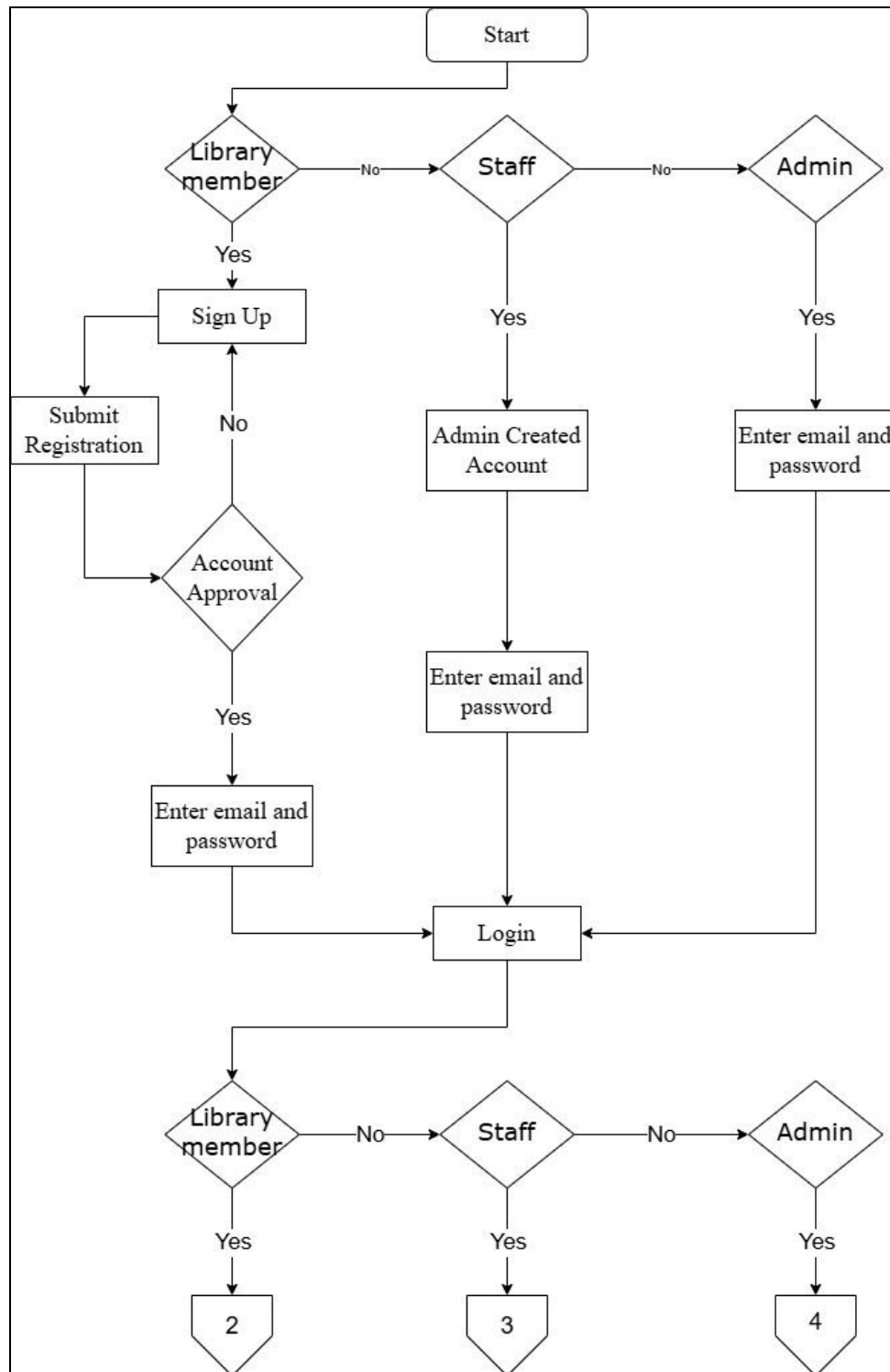


Fig. 4 Login flowchart

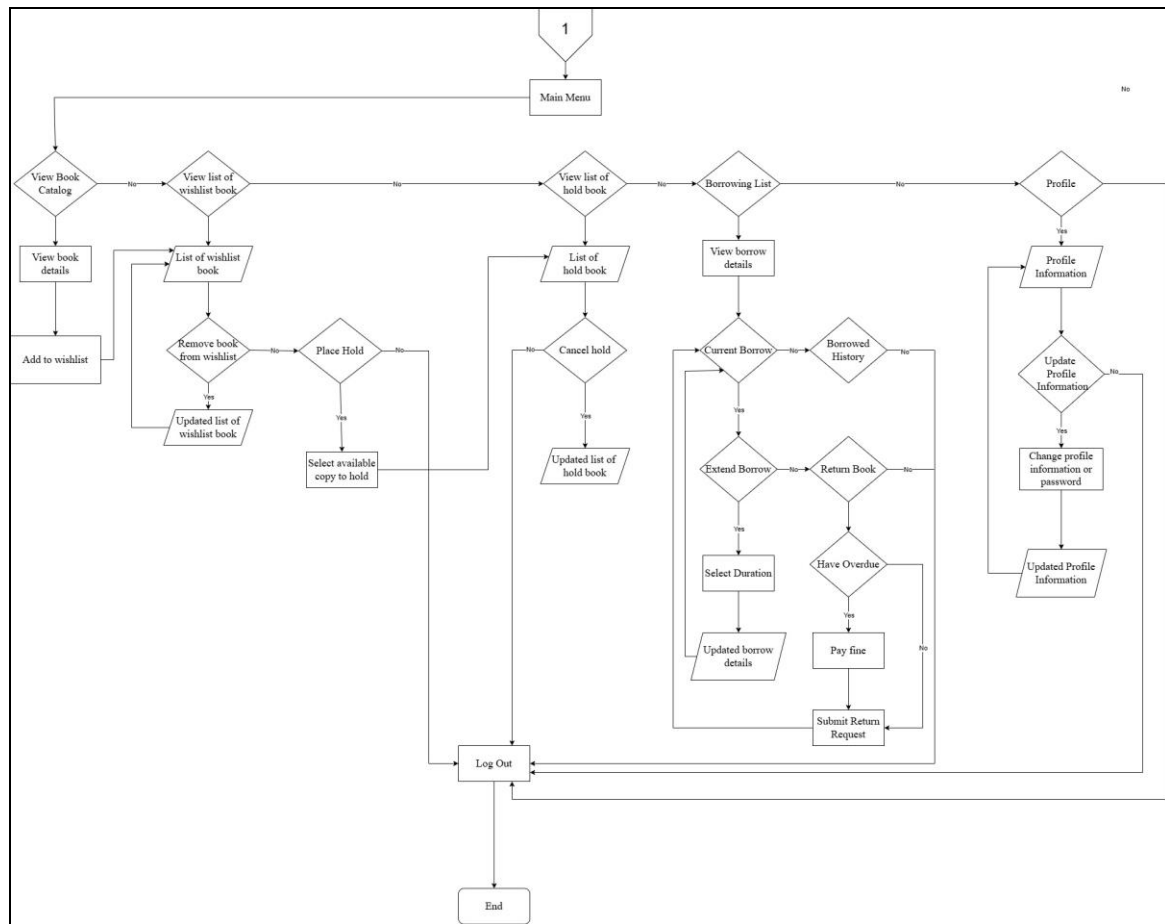


Fig. 5 Library member flowchart

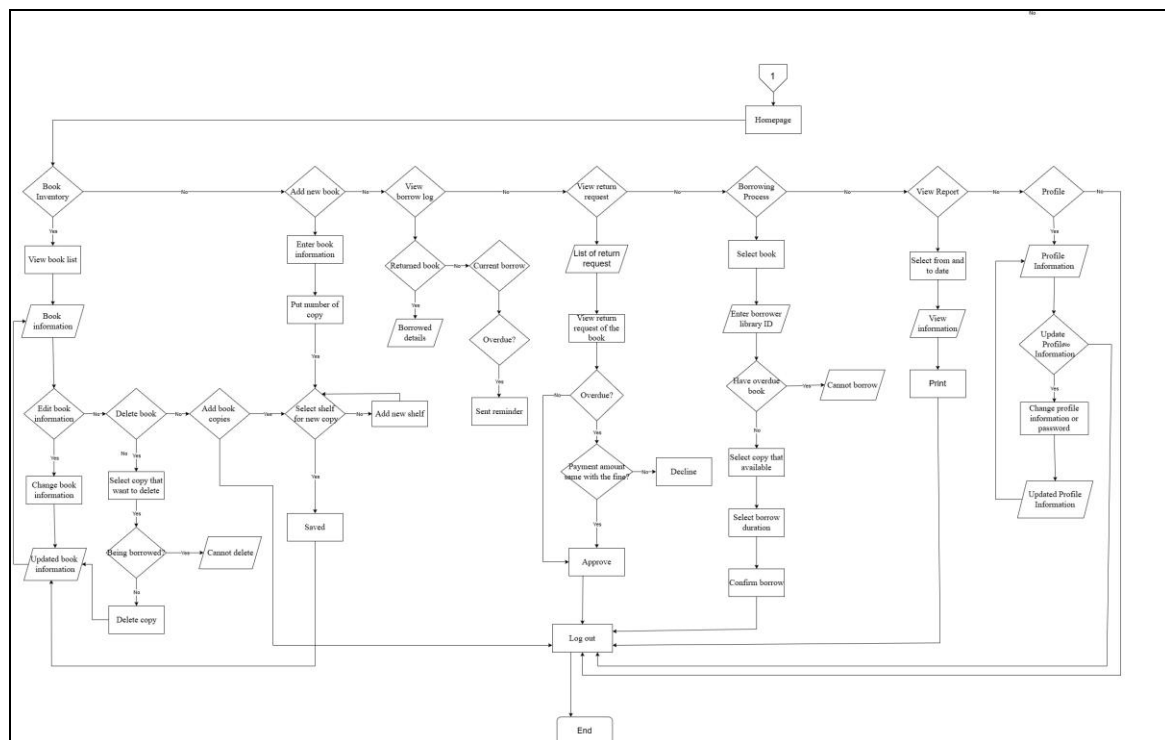


Fig. 6 Staff flowchart

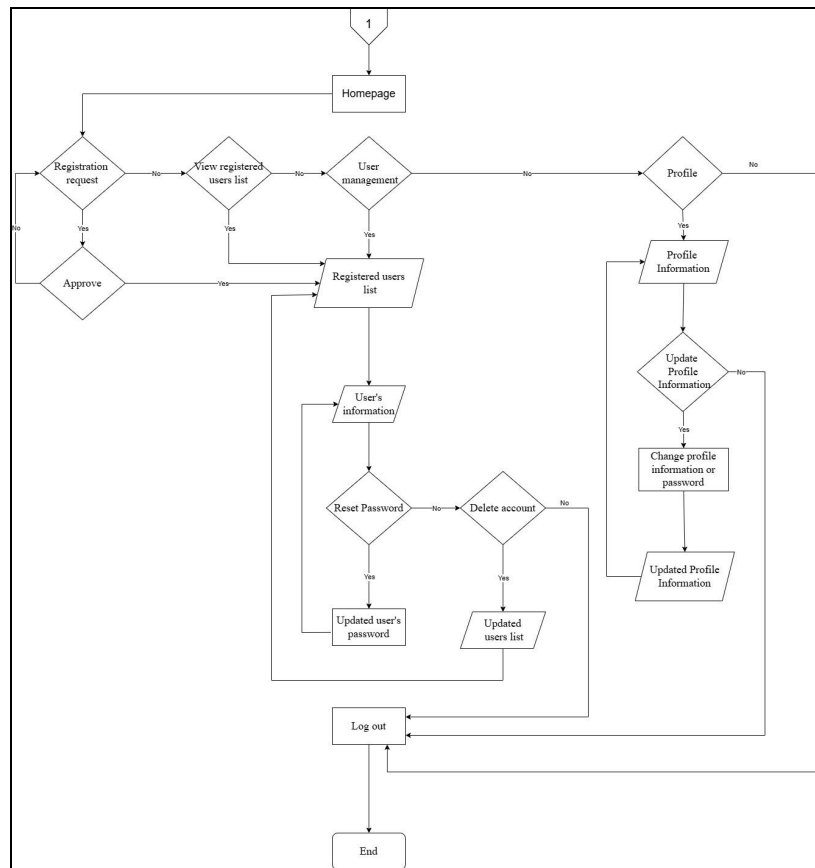


Fig. 7 Admin flowchart

4.3 Interface Design

The design phase is a critical step in system development, focusing on translating requirements into a structured blueprint for implementation. During this phase, the interface for the Library Management System were carefully designed to visualize the system's functionality and structure. This step was crucial in ensuring a clear understanding of the system before moving on to the coding and implementation phase. Figures 8 to 11 showcase the system's user interface, highlighting key screens and their design elements.

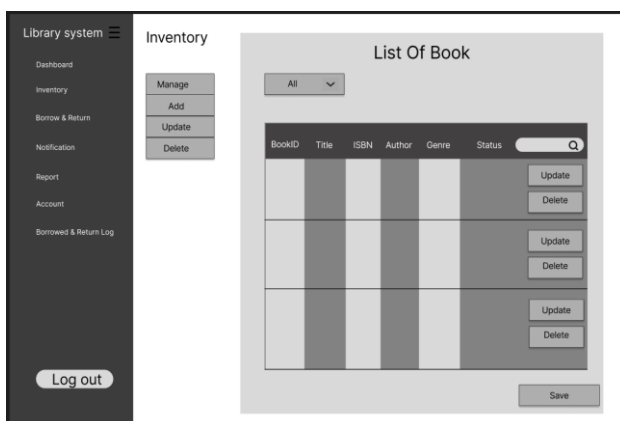


Fig. 8 Inventory Page

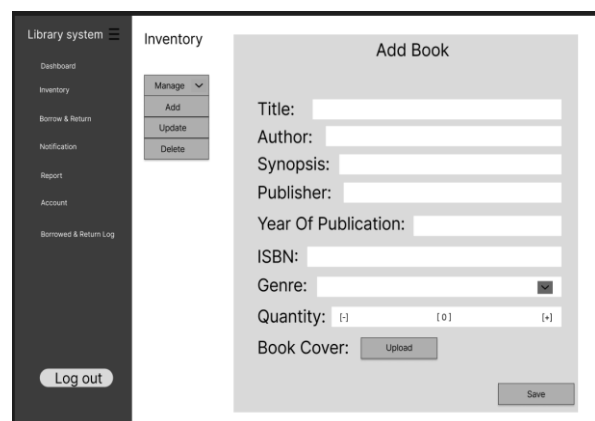


Fig. 9 Add Book Page

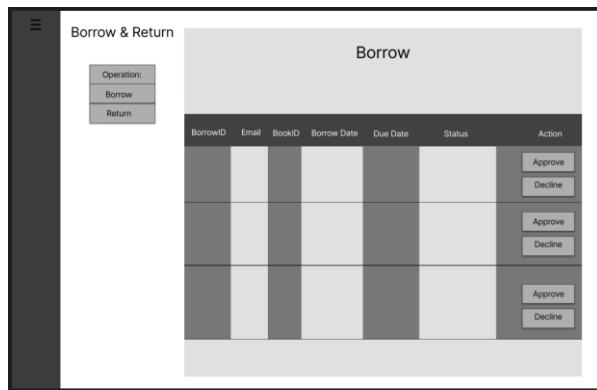


Fig. 10 Borrow Request Page

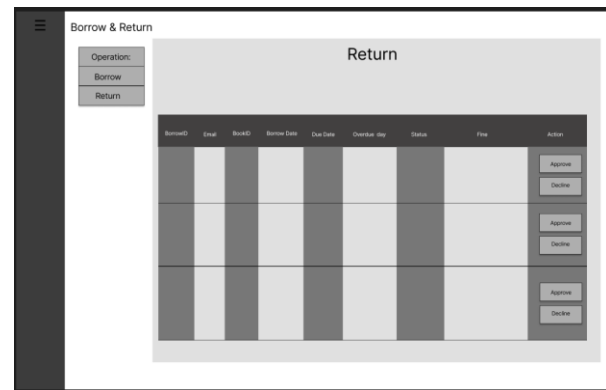


Fig. 11 Return Approval Page

5. Implementation

The Library Management System is implemented through several functional modules. Each module is designed to handle a specific part of the system's functionality, ensuring a clear separation of duties for Library Members, Staff, and Admins.

5.1 Login Interface

Figure 11 shows the login interface where users enter their username and password to access the system. Figure 12 displays the source code that handles the login process, including credential verification and redirection after successful login.

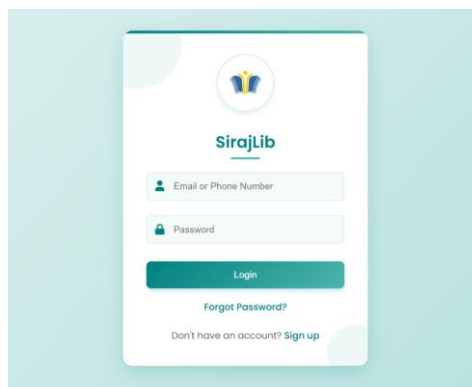


Fig. 12 Login interface

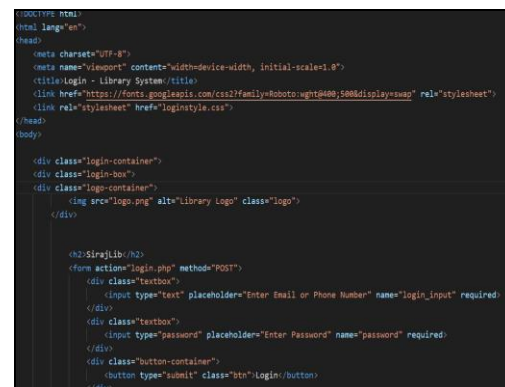


Fig. 13 Login source code

Figure 13 shows the registration interface where new users can create an account by providing their name, email, username, and password. The form ensures that all required fields are filled in correctly before submission. Figure 14 displays the source code that handles the registration process, including input validation, password encryption, and inserting the new user data into the database. A success message is shown once registration is complete.

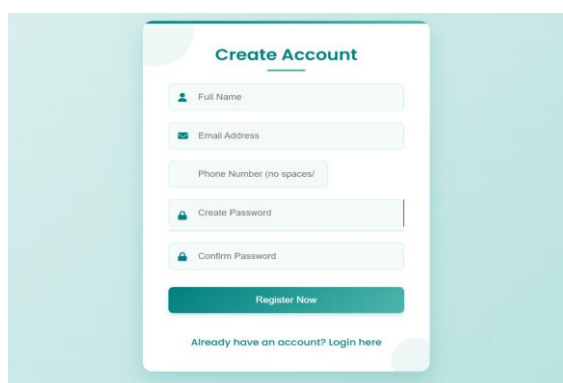


Fig. 14 Register interface



Fig. 15 Register source code

Figure 15 shows the book catalog interface, where users can browse and search for books available in the library. The catalog displays key information such as book title, author, category, and availability status. Figure 16 presents the PHP source code that retrieves book data from the database and dynamically displays it on the catalog page. Users can use the search bar to filter results and view more details about a selected book.

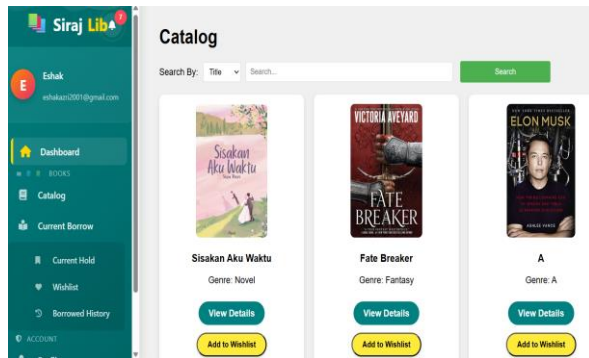


Fig. 16 Book catalog interface

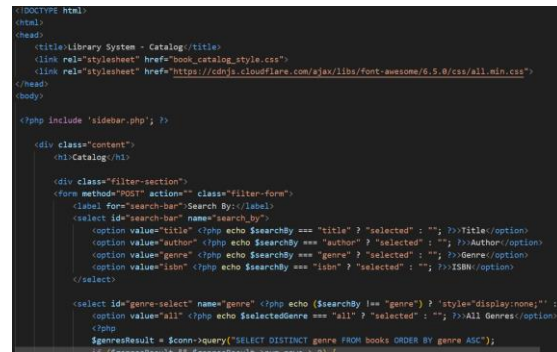


Fig. 17 Catalog source code

Figure 17 shows the return request interface available to library members, where they can submit a return request. The interface also displays any overdue fines, if applicable. If a fine is present, members can make a payment by scanning the provided QR code. Figure 18 displays the source code that handles the return request process, updates the book status in the database, calculates any overdue fines, and records the payment once completed.

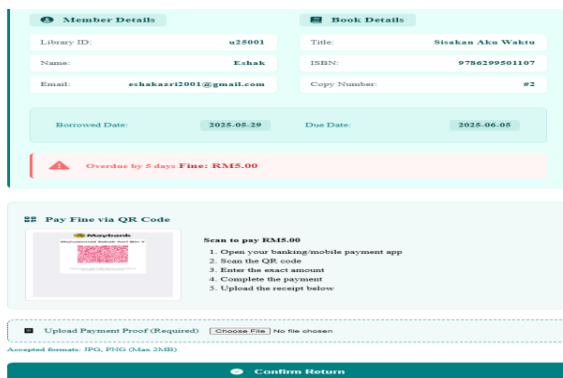


Fig. 18 Return request interface

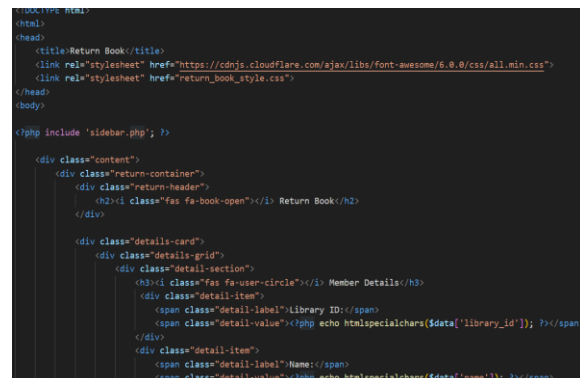


Fig. 19 return request source code

Figure 19 shows the interface of the extend borrow page. The user can choose between extending the loan by 1 week or 2 weeks. Once the user selects the desired duration and submits the form, the new return date is updated accordingly in the system. Figure 20 displays the source code that handles the extension process. It captures the user's selection, validates the request, and updates the borrow record in the database with the newly calculated return date.

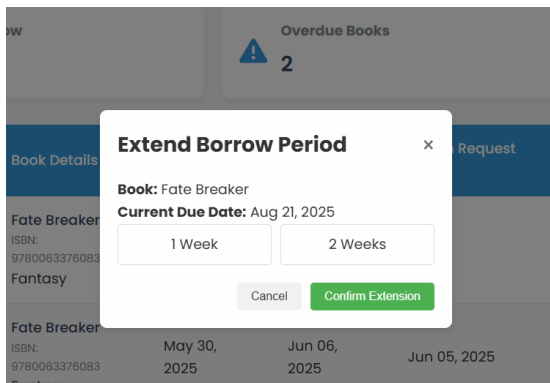


Fig. 20 Extend borrow interface

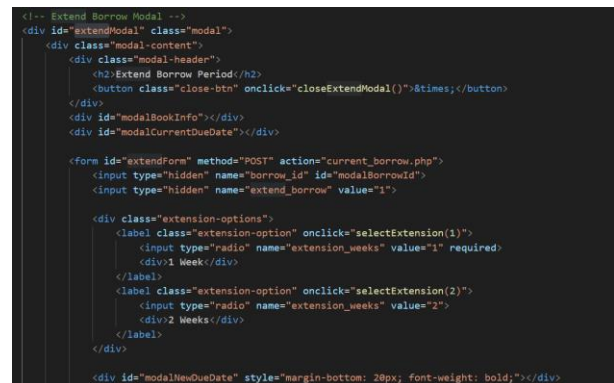


Fig. 21 Extend borrow source code

Figure 21 shows the interface for adding books. This screen presents fields for entering all the necessary book details, followed by options to specify the quantity and choose a shelf location. This design streamlines the process by allowing for the selection of an existing shelf or the quick addition of a new one as needed. Figure 20 displays the source code. This code manages how the system processes all your input, from the book information and quantity to handling shelf selection and enabling the creation of new shelves. It ensures a seamless and efficient experience for cataloging new books.

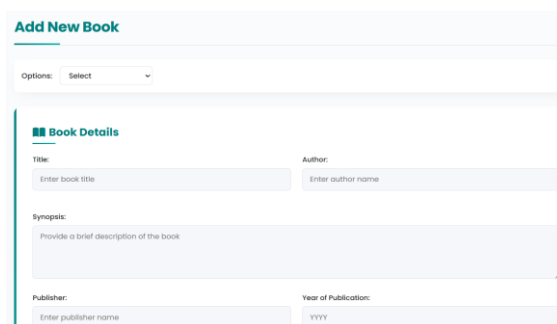


Fig. 22 Add new borrow interface

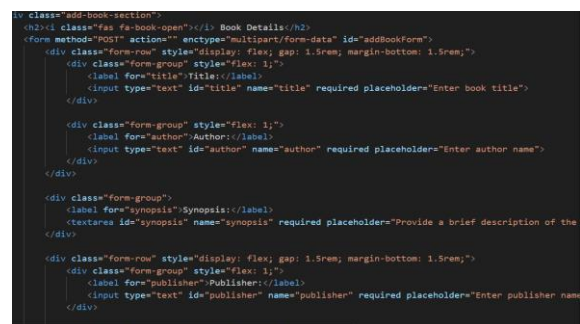


Fig. 23 Add new book source code

Figure 23 shows the interface of the "Borrow Book" form used by staff. In this form, staff are required to enter the user's Library ID, select an available copy of the book from a dropdown list, and choose the loan duration. Figure 24 displays the source code that powers the borrow form. It includes input handling for the user's Library ID, dynamic selection of available book copies, and loan duration options. The code also manages database interactions to record the borrowing transaction and update the book copy's status to "borrowed."

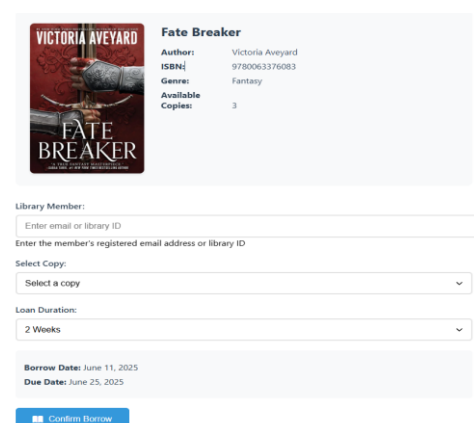


Fig. 24 Add new borrow interface

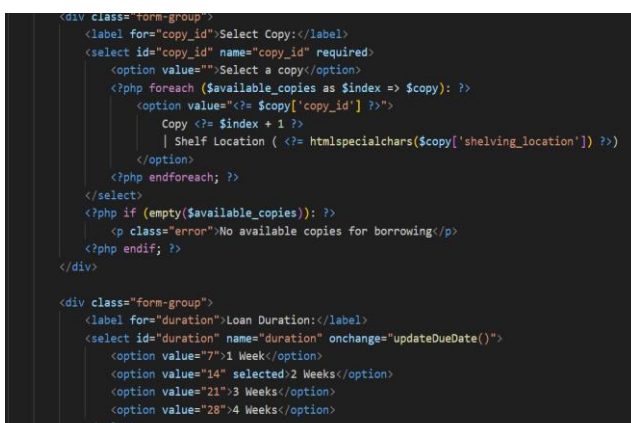


Fig. 25 Add new book source code

Figure 25 shows the report interface used by staff to view borrowing information. It displays details such as total transactions, active borrows, overdue items, most borrowed book, most active user, return rate, and borrowing trends by genre. The layout is clear and easy to understand. Figure 26 shows the source code for the report page. The code gets data from the database, calculates important values, and shows them on the page. It also creates charts to show borrowing patterns and helps highlight top users and books.

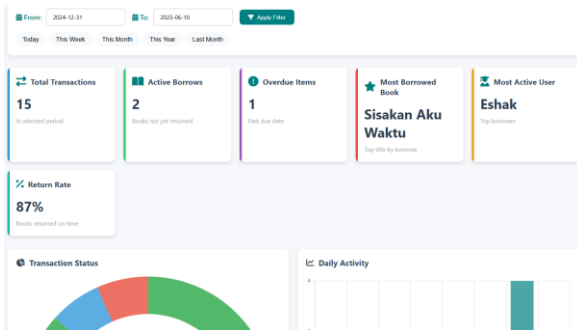


Fig. 26 Add new borrow interface

```
<div class="stat-grid">
  <div class="stat-card">
    <h3><i class="fas fa-exchange-alt stat-icon"></i> Total Transactions</h3>
    <div class="stat-value"><php echo count($reportData); ?></div>
    <p class="stat-description">In selected period</p>
  </div>

  <div class="stat-card">
    <h3><i class="fas fa-book-open stat-icon"></i> Active Borrows</h3>
    <div class="stat-value">
      <php
        $activeBorrows = array_filter($reportData, fn($item) => $item['return_date'] == null);
        echo count($activeBorrows);
      >
    </div>
    <p class="stat-description">Books not yet returned</p>
  </div>

  <div class="stat-card">
    <h3><i class="fas fa-exclamation-circle stat-icon"></i> Overdue Items</h3>
    <div class="stat-value">
      <php
        $overdueItems = array_filter($reportData, fn($item) =>
          $item['return_date'] == null && strtotime($item['due_date']) < time());
        echo count($overdueItems);
      >
    </div>
  </div>
</div>
```

Fig. 27 Add new book source code

6. Result and Discussion

This section presents the testing conducted on the developed system. The testing process involved two main parts: system testing and user feedback. The aim was to ensure that all features functioned correctly, met the specified project requirements, and maintained proper security. All modules were tested to confirm that the system is stable and ready for actual use.

6.1 Testing Result

Table 5 Testing Result

Module	Description	Expected Result	Actual Result	Pass/Fail
Book Management	To test whether the staff can add, update, and delete book records.	Staff is able to add, update, and delete books in the system.	Staff successfully added, updated, and deleted book records.	Pass
Login and Registration	To test whether users and admin can register, and admin can create staff accounts.	Users can register and update their profiles. Only admin can create staff accounts.	Users successfully registered and admin created staff accounts.	Pass
Borrow and Return	To test whether staff can process borrowing and return, and users can view borrowing history.	Staff can process borrow/return books, and users can view their borrowing details and history.	Borrowing and return were processed correctly. Users were able to view their borrowing records.	Pass
Book Searching	To test whether users can search for books and see their availability status.	Users can search for books, and see each book's availability status ("Available"/"Not Available") is shown.	Users successfully searched for books, and the availability status was correctly displayed.	Pass
Notification Module	To test whether the system sends overdue notifications.	Overdue book reminders are sent to users via email.	Notifications for overdue books appeared correctly on the website.	Pass
Reporting	To test whether the system generates detailed transaction reports.	A detailed report of all borrow and return transactions is generated for staff viewing.	Detailed transaction reports were generated successfully and accurately.	Pass

6.2 User Acceptance Testing

This section presents the analysis of user feedback from 10 respondents regarding the Library Management System. Each figure displays a pie chart summarizing user responses to a specific evaluation question.

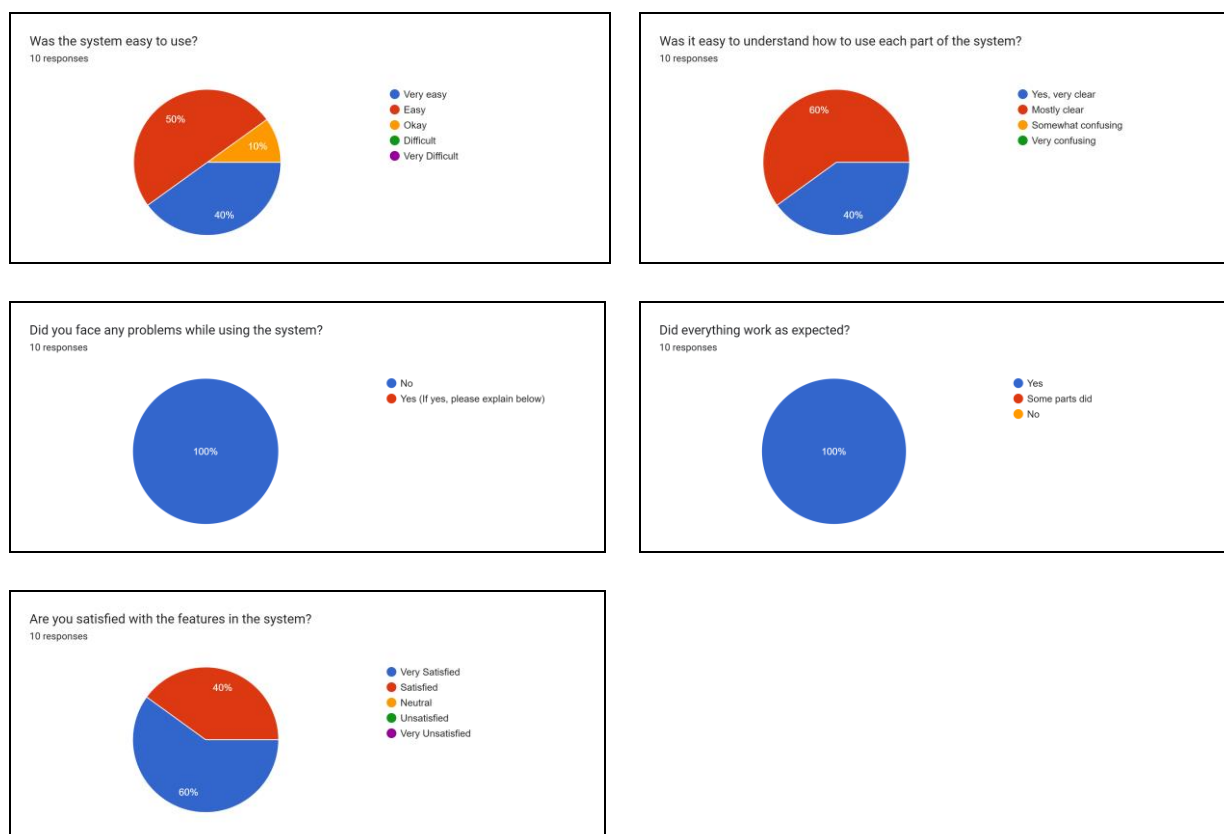


Fig. 27 Pie Chart of Testing Result for System Functionality

Figure 27 presents a series of pie charts summarizing feedback from 10 participants during the User Acceptance Testing (UAT) of the Library Management System. The results indicate that 90% of respondents found the system either easy or very easy to use, while all participants agreed it was clear to understand, with 40% selecting "very clear" and 60% "mostly clear." In addition, 100% reported no issues while using the system and confirmed that all functions worked as expected. Regarding overall satisfaction, all users expressed positive feedback, with 60% being very satisfied and 40% satisfied with the features provided. These results suggest that the system performs well, is easy to operate, and meets user needs effectively.

7. Conclusion

In conclusion, the proposed system has been successfully developed according to the project goals. All key modules including book management, login and registration, borrowing and returning, book searching, notifications, and reporting were fully implemented and tested. The system performed well during testing, with all functions working smoothly and meeting the specified requirements. User feedback also showed that the system is straightforward, easy to use, and provides the necessary features. With no major issues found, the system is considered stable and ready to be used in a real setting to support library operations. One limitation of the system is that fine payments must be made manually, which may slow down the return process and require additional verification by staff. For future work, the system can be improved by introducing digital library access, allowing users to read e-books and documents online. Additionally, the system can be enhanced with barcode or QR code scanning features to speed up book borrowing and returning processes, making transactions more efficient for both staff and members.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

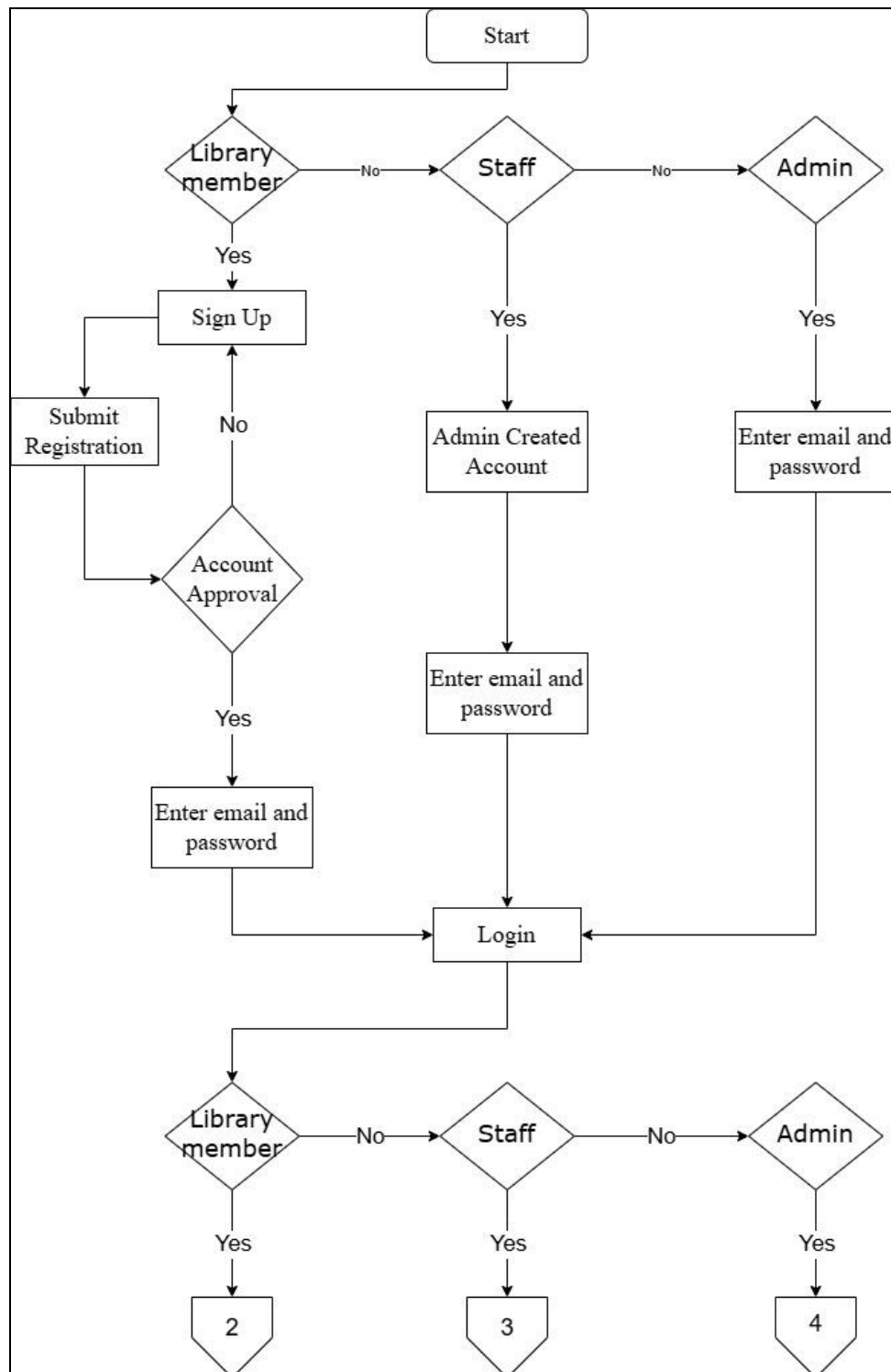
Author Contribution

*The authors confirm contribution to the paper as follows: **study conception and design:** Muhammad Eshak Azri, Rozita Abdul Jalil; **data collection:** Muhammad Eshak Azri; **analysis and interpretation of results:** Muhammad Eshak Azri, Rozita Abdul Jalil; **draft manuscript preparation:** Muhammad Eshak Azri, Rozita Abdul Jalil. All authors reviewed the results and approved the final version of the manuscript.*

References

- [1] "Tunku Tun Aminah Library, UTHM," *ptta.uthm.edu.my*. <https://ptta.uthm.edu.my/>
- [2] "Official Website of Koha Library Software," *Koha-community.org*, 2019. <https://koha-community.org/>
- [3] "Home |," *umlib.um.edu.my*. <https://umlib.um.edu.my/>
- [4] E. B. Ayo, R. N. Jotic, A. Raqueño, J. V. G. Loresca, I. F. Mendoza, and P. V. M. Baroña, "Development of an Integrated Library Management System (ILMS)," *International Journal of Interactive Mobile Technologies (ijim)*, vol. 17, no. 10, pp. 242–256, May 2023, doi: <https://doi.org/10.3991/ijim.v17i10.37509>.
- [5] Glavin Gaga, "The Digital Library Management System 2021 : Design and Implementation of an integrated Library Management System," Feb. 05, 2021. https://www.researchgate.net/publication/369020218_The_Digital_Library_Management_System_2021_Design_and_Implementation_of_an_integrated_Library_Management_System
- [6] Ali Hadi Abdulwahid, Manjula Pattnaik, M. R. Palav, G. Tilak, G. Manoharan, and G. Pandi Selvi, "Library Management System Using Artificial Intelligence," Apr. 2023, doi: <https://doi.org/10.1109/iconstem56934.2023.10142904>.
- [7] W.-Y. Chiu, W. Meng, and W. Li, "LibBlock - Towards Decentralized Library System based on Blockchain and IPFS," Dec. 2021, doi: <https://doi.org/10.1109/pst52912.2021.9647821>.
- [8] Y. Gao, "Construction of Smart Library Service System Based on Internet of Things Data Technology," *IEEE Xplore*, Dec. 01, 2022. <https://ieeexplore.ieee.org/document/10059734>
- [9] Z. Li, "Construction of an Intelligent Library Book Management System Based on Artificial Intelligence Algorithms," Sep. 2023, doi: <https://doi.org/10.1109/peeec60561.2023.00135>.
- [10] Andrew Jerico Budiarto, N. Wijaya, R. Heriawan, Felix Indra Kurniadi, and Budi Juarto, "Design and Implementation of a Web-Based Library Management System," Oct. 2023, doi: <https://doi.org/10.1109/iccteie60099.2023.10366726>

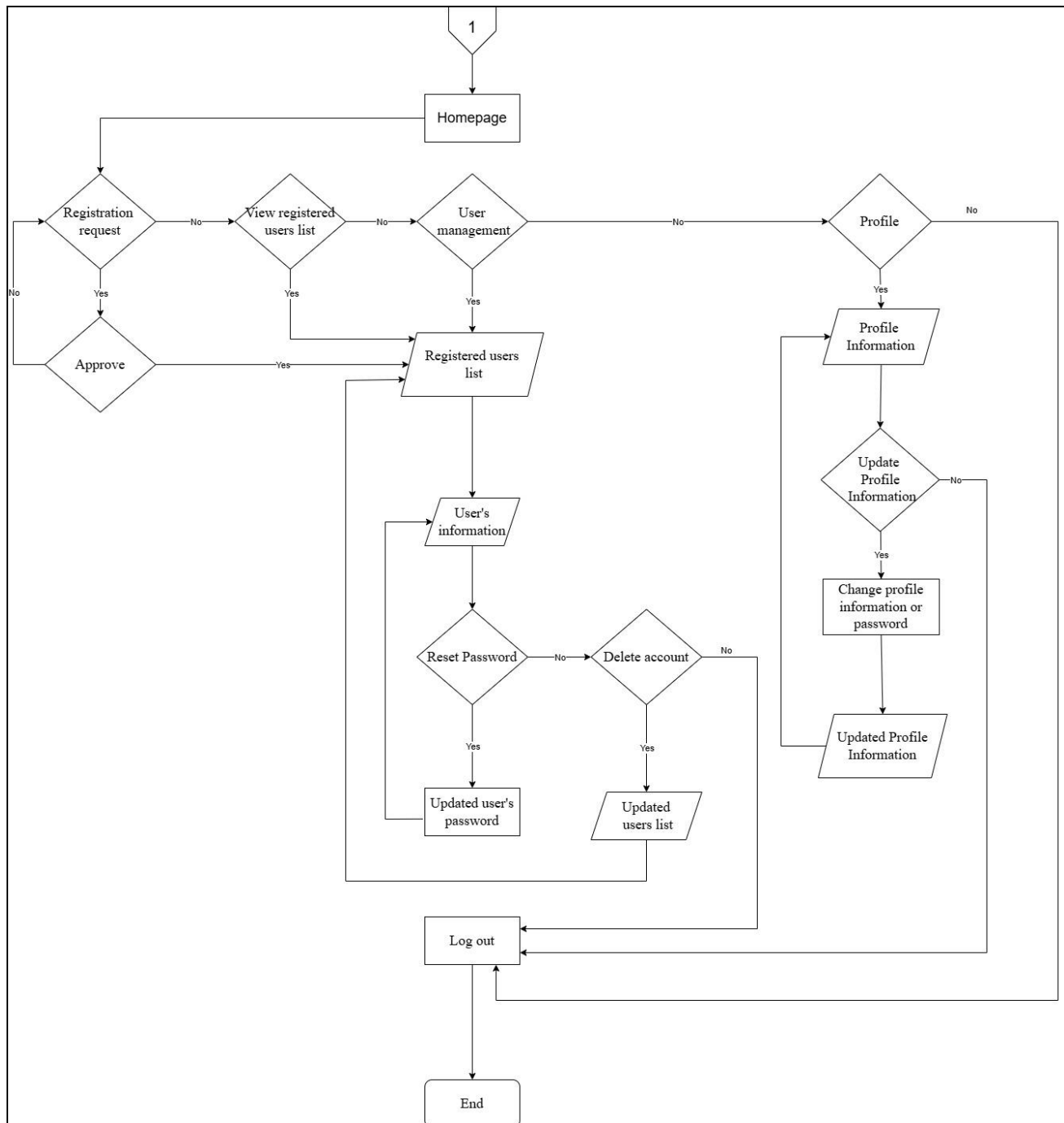
Appendix A System's Flowchart



Login flowchart

Library member flowchart

Staff flowchart



Admin flowchart