MINISTRY OF EDUCATION AND TRAINING

UNIVERSITY OF ECONOMICS AND FINANCE

Subject: Java Technology

MID-TERM PROJECT REPORT

TOPIC:	**Library Management System**
Class:	**242.ITE1219E.B06E**
Instructor:	**MSc. Hoàng Văn Hiếu**

Students:

1.	**Võ Đăng Khoa**	**SID: 225052036**
2.	**Dương Thị Thanh Thảo**	**SID: 225053167**
3.	**Lê Thiện Nhân**	**SID: 225053134**
4.	**Trần Thanh Độ**	**SID: 225051852**

HO CHI MINH CITY 2024 – 2025

Table of Contents

List of Figures

List of Pictures

Revision History

Name	**Date**	**Reason For Changes**	**Version**
	12/6/2025	initial version	Midterm
		baseline following changes after inspection	Final

INTRODUCTION

The digital transformation of educational institutions has fundamentally redefined operational standards, with library management representing a critical area requiring technological intervention. Contemporary academic libraries face unprecedented challenges in balancing traditional service delivery with evolving user expectations for seamless, technology-enabled access to information resources.

This report presents the development of a comprehensive Library Management System designed to address operational challenges in modern academic library environments. The project strategically applies Java technology and object-oriented programming principles to create an integrated digital platform that transforms traditional library workflows while maintaining service quality standards. The system architecture addresses critical deficiencies through automated book cataloging, streamlined member management, and intelligent lending processes that eliminate manual documentation requirements while enhancing data accuracy and accessibility.

Through systematic analysis of library management requirements and strategic technology integration, this project establishes a foundation for future enhancements while delivering immediate operational improvements. The development process emphasizes adherence to software engineering principles while maintaining flexibility to accommodate unique operational requirements of library management systems.

Chapter 1: Overview

1.1 Research Requirements and Topic Selection Rationale

The rapid digitization of educational institutions has fundamentally transformed operational expectations, creating an urgent need for modern library management solutions that can effectively address the complexities of contemporary academic environments. Traditional library management systems, characterized by manual record-keeping processes and paper-based cataloging methods, have proven inadequate in meeting the demands of today's technology-driven educational landscape. The inefficiencies inherent in these legacy systems manifest through frequent data inconsistencies, time-consuming administrative procedures, and limited accessibility that ultimately compromise both operational effectiveness and user satisfaction.

The selection of this research topic stems from extensive analysis of current library management challenges observed across academic institutions, where the gap between technological capabilities and operational implementation has become increasingly pronounced. Manual inventory tracking systems consistently demonstrate vulnerability to human error, while the absence of integrated digital workflows creates significant barriers to efficient resource management. These operational deficiencies directly impact the quality of educational support services and limit the strategic value that modern library systems should provide to academic communities.

The compelling nature of this research lies in its potential to demonstrate how contemporary Java technology can be strategically applied to solve real-world operational challenges while establishing a foundation for future technological advancement. The development of a comprehensive Library Management System using Java represents an opportunity to bridge theoretical software engineering knowledge with practical application development, creating tangible value for educational institutions while advancing understanding of enterprise software development methodologies.

1.2 Research Objectives, Target Audience, and Scope

The primary objective of this research is to develop a robust Library Management System that fundamentally transforms traditional library operations through the strategic implementation of Java technology and modern software engineering principles. This objective encompasses the creation of an integrated digital platform that automates core library functions including book cataloging, member management, and circulation processes while maintaining the highest standards of data integrity and user experience design.

The research specifically targets academic library environments where operational efficiency directly impacts educational quality and resource accessibility. The intended audience includes library administrators who require comprehensive oversight capabilities for strategic decision-making, library staff members who manage daily operational tasks including circulation and cataloging functions, and academic researchers interested in practical applications of Java technology in institutional management systems. The system design accommodates varying levels of technological proficiency while ensuring that all stakeholders can effectively utilize system capabilities relevant to their specific operational responsibilities.

The scope of this research encompasses the complete development lifecycle of a desktop-based Library Management System, including comprehensive requirements analysis, system architecture design, database implementation, user interface development, and thorough testing procedures. The research addresses fundamental library operations including inventory management, patron services, lending and return processes, and administrative reporting functions. However, the current scope deliberately excludes advanced features such as interlibrary loan management, digital resource integration, and web-based accessibility to maintain focus on core functionality development while establishing a solid foundation for future system enhancements.

1.3 Report Structure

The comprehensive analysis and development process documented in this report is organized according to a logical progression that guides readers through theoretical foundations, practical implementation, and research conclusions. The structured approach ensures thorough coverage of all essential aspects while maintaining clarity and coherence throughout the technical documentation.

Chapter 1: Overview provides essential context through examination of research requirements and topic selection rationale, establishment of clear research objectives and target audience identification, and presentation of the comprehensive report structure that guides subsequent technical analysis.

Chapter 2: Theoretical Foundation establishes the conceptual framework underlying the system development process through detailed exploration of software engineering methodologies, comprehensive analysis of object-oriented programming principles, examination of Java technology stack capabilities, investigation of database management system requirements, exploration of user interface design principles, and thorough coverage of software testing methodologies essential for quality assurance.

Chapter 3: System Architecture presents the complete technical implementation framework through systematic system requirements analysis encompassing both functional and non-functional specifications, comprehensive system design documentation including architectural overview and detailed design diagrams, user interface design principles with implementation examples, database design specifications with schema documentation, and system implementation details covering technology stack selection and code organization strategies.

Chapter 4: Conclusion and Development Directions synthesizes research outcomes through comprehensive project summary highlighting key achievements and deliverables, critical analysis of challenges encountered and solutions implemented, reflection on lessons learned throughout the development process, identification of future enhancement opportunities, and presentation of strategic recommendations for system deployment and ongoing maintenance.

1.4 Project Scope and Limitations

The scope of this Library Management System encompasses the development of a comprehensive desktop application that addresses the core operational requirements of academic library environments. The system will provide complete book management functionality, including the ability to add new resources to the collection, modify existing records to reflect changes in cataloging information, and remove outdated or damaged materials from the active inventory. Member management capabilities will support patron registration, profile maintenance, and borrowing privilege administration.

The lending and return operations module will automate transaction processing while maintaining detailed historical records for accountability and analytical purposes. Inventory tracking functionality will provide real-time visibility into resource availability and location status, enabling staff to respond effectively to patron inquiries and optimize collection organization. The system will incorporate comprehensive reporting capabilities that generate statistical summaries, utilization analyses, and operational performance metrics.

However, the current project scope is constrained by several important limitations that define the boundaries of the initial implementation. The system is designed as a single-user desktop application, which restricts concurrent access capabilities and limits deployment to individual workstations rather than networked environments. The functional scope is deliberately focused on core library operations, excluding advanced features such as interlibrary loan management, digital resource integration, or sophisticated acquisition workflows.

The system operates as a standalone application without web-based access capabilities, which limits remote accessibility and mobile device compatibility. Additionally, the current scope does not include integration with external library systems, cataloging standards, or third-party databases, which may be desirable for institutions operating within broader library consortiums or requiring compliance with specific metadata standards.

1.5 Target Users

The Library Management System is specifically designed to serve the operational needs of library personnel who are responsible for daily administrative functions within academic library environments. The primary user category consists of library administrators who require comprehensive oversight capabilities for collection management, patron services, and operational reporting. These users typically possess moderate to advanced computer literacy and are responsible for strategic decision-making regarding library operations and resource allocation.

Library staff members represent another critical user category, encompassing circulation desk personnel, cataloging specialists, and reference librarians who interact directly with the system during routine patron service activities. These users require intuitive interfaces that facilitate efficient transaction processing while maintaining accuracy in record-keeping and patron interaction documentation. The system accommodates varying levels of technological proficiency within this user group through streamlined workflows and clear navigational structures.

Librarians responsible for specialized functions such as collection development, acquisitions, and technical services constitute an additional user category with specific requirements for analytical reporting and inventory management capabilities. These users benefit from the system's comprehensive data management features and reporting functionality that supports evidence-based decision-making in collection development and resource optimization strategies.

The system architecture recognizes that library environments often operate with limited staffing resources and varying levels of technological infrastructure. Consequently, the user interface design prioritizes accessibility and efficiency to ensure that all designated user categories can effectively utilize system capabilities without requiring extensive training or technical support.

1.6 Development Environment and Tools

The development of this Online Library Management System was carried out using a modern and robust technology stack, chosen to ensure reliability, scalability, and ease of maintenance.

The backend is built on the **Java** programming language, utilizing the **Spring Boot** framework to streamline development and create a standalone web application. For the user interface, the project is a web application that employs standard web technologies: **HTML**, **CSS**, and **JavaScript**, with **Thymeleaf** serving as the server-side template engine for rendering dynamic content, and **Bootstrap** for creating a responsive design.

The primary tools and environment used in the project are as follows:

- **IDE: IntelliJ IDEA** was used as the main Integrated Development Environment for coding, debugging, and project management.
- **Database: Microsoft SQL Server** was chosen as the relational database management system to store and manage all application data.
- **Build Tool: Maven** was used for managing project dependencies and automating the build process.
- **Version Control: Git** was implemented for source code management, tracking changes, and facilitating collaboration.
- **Web Server:** The application runs on an embedded **Apache Tomcat** server, provided by default with Spring Boot, simplifying deployment.

Chapter 2: Theoretical Foundation

2.1. Introduction

This chapter presents the technological platforms, architecture, and theoretical foundations applied to build and develop the Online Library Management System. The selection of these technologies aims to ensure the system has good performance, is secure, easy to maintain, and scalable in the future. The core technologies include the Spring Framework, Microsoft SQL Server, frontend technologies, and project management tools.

2.2. Backend (Server-side) Technologies

The core of the system is built on the Java platform, with strong support from the Spring Framework to handle business logic, data management, and security.

2.2.1. Spring Framework

Spring is a popular and powerful Java application development framework that provides a comprehensive programming and configuration model. In this project, the main modules of Spring were utilized:

+ Spring Boot was used to simplify the setup and deployment process by providing smart default configurations, an embedded Tomcat web server, and eliminating much of the complex XML configuration, allowing the development team to focus on writing business logic.

+ The project's foundation is the Spring MVC (Model-View-Controller) architecture, which clearly separates the components for handling logic, managing data, and the user interface. The Model consists of Java objects (POJOs) like User, Notification, and Book. The View component, primarily built with Thymeleaf and HTML/CSS, is responsible for rendering dynamic data from the server. The Controllers are Java classes, such as AuthController, HomeController, and NotificationController, that handle user requests, interact with Service classes for business processing, and return the appropriate Model and View.

For database interaction, the project uses Spring Data JPA, a module that simplifies the construction of data access layers. It operates based on the Java Persistence API (JPA) and uses Hibernate as the ORM (Object-Relational Mapping) implementation provider, which allows for the automatic mapping of Java objects (@Entity) to database tables. Spring Data JPA also provides a repository mechanism, where interfaces like UserRepository extend JpaRepository to automatically provide basic CRUD methods without needing to write SQL code. To ensure system security, Spring Security is integrated to handle authentication and authorization. It checks user identity upon login, using a PasswordEncoder to encrypt passwords before storage. Authorization is managed by controlling user access to resources based on their roles (READER, STAFF, ADMIN), enforced via annotations like @PreAuthorize. The core principle of Spring, Dependency Injection (DI) and Inversion of Control (IoC), is also heavily used. The Spring Container manages the creation and injection of dependencies, as seen with the @Autowired annotation in controller classes to inject Service objects.

2.3. Frontend (Client-side) Technologies

The user interface is built with standard web technologies to ensure interactivity, user-friendliness, and compatibility across various devices. The foundational trio of HTML5, CSS3, and JavaScript is used to build the structure, format the interface, and create client-side interactive behaviors. Thymeleaf, a server-side Java template engine, is seamlessly integrated with Spring MVC to insert dynamic data from the Model directly into HTML files, simplifying the display of information like book lists and user profiles. Bootstrap, a popular CSS/JavaScript framework, is used to quickly build a responsive user interface, providing a grid system and pre-built components that ensure the website displays well on both desktops and mobile devices.

2.4. Database Management System

Microsoft SQL Server was chosen as the relational database management system (RDBMS) to store all application data. SQL Server offers stable performance, strong security features, and the ability to manage large volumes of data, making it suitable for the requirements of a library system.

2.5. Tools and Environment

IntelliJ IDEA served as the primary Integrated Development Environment (IDE) for writing code, debugging, and managing the project, offering powerful support tools for Spring Boot development that increased productivity.

The application is served by Apache Tomcat, which is used as the web server and servlet container; it comes conveniently embedded with Spring Boot, making deployment straightforward. Maven is utilized as the project management and dependency management tool, automatically downloading necessary libraries and managing the build process. For version control, Git was used as the distributed version control system to track source code changes, facilitating teamwork and managing project versions.

2.6. Design Patterns and Diagrams

The MVC (Model-View-Controller) pattern is the main architectural design, which helps in separating the application's concerns, thereby increasing modularity and maintainability. Before programming, the Unified Modeling Language (UML) was used to analyze and design the system. UML diagrams such as the Class Diagram were used to describe the static structure of the system, including classes and their relationships. The Sequence Diagram was used to illustrate the interactions between objects over time for key functions like login and borrowing books. Finally, the State Diagram was employed to describe the various states of an object throughout its lifecycle, such as the status of a Book.

2.7 Software Engineering Methodologies

The development of the Library Management System adheres to established software engineering methodologies that emphasize systematic approaches to software development, quality assurance, and project management. The project implementation draws primarily from iterative development principles, which facilitate continuous refinement of system components through cyclical development phases that incorporate feedback mechanisms and progressive enhancement strategies.

The iterative methodology proves particularly advantageous for educational software projects where requirements may evolve based on user feedback and technical discoveries during the development process. This approach enables the development team to establish foundational system architecture while maintaining flexibility to accommodate modifications and improvements identified through testing and evaluation phases. The methodology emphasizes incremental delivery of functional components, allowing for early validation of core concepts and identification of potential design challenges before they become embedded in the system architecture.

Agile principles inform the project's approach to collaborative development and adaptive planning, particularly in the context of small-team dynamics characteristic of academic software projects.

The emphasis on working software over comprehensive documentation aligns with the practical constraints of educational timelines while ensuring that functional deliverables remain the primary focus of development efforts. However, the academic context necessitates maintaining comprehensive documentation standards to support learning objectives and project evaluation requirements.

The software engineering approach incorporates risk management strategies that identify potential development challenges early in the project lifecycle and implement mitigation strategies to ensure project success. Configuration management principles guide version control practices and change management procedures, ensuring that all modifications to the system are properly documented, tested, and integrated into the overall project architecture.

Chapter 3: System Architecture

3.1 System Requirements Analysis

3.1.1 Functional Requirements

The library management system encompasses comprehensive functionality across three primary user categories, each with distinct operational requirements and access privileges.

For library members, the system provides essential **account management** capabilities that enable users to register new accounts, authenticate through secure login procedures, and maintain session control through logout functionality. The **search functionality** represents a core component, allowing members to locate books efficiently through multiple search parameters including title, author, genre classification, and ISBN identification. The **borrowing and returning process** facilitates seamless transaction management, enabling users to submit borrowing requests and manage their return obligations effectively. Additionally, **personal information management** empowers users to maintain current profile information and access comprehensive borrowing history records.

Library staff operations center on advanced administrative capabilities that extend beyond basic member functions. **Book inventory management** provides comprehensive control over the library catalog, including the addition of new titles, modification of existing records, and removal of outdated materials. **Borrowing and returning administration** grants staff members the authority to approve or decline borrowing requests while managing the complete return process. **Inventory monitoring** ensures accurate tracking of book conditions, including identification and documentation of damaged or lost materials. The system also incorporates **reporting functionality** that generates detailed reports on borrowing patterns, return statistics, and overall inventory status. Administrative personnel possess the highest level of system access, encompassing **user account administration** for both library members and staff accounts. **Category management** functionality allows administrators to create, modify, and remove book classification systems to maintain organized catalog structures. **System administration** capabilities include updating library policies, managing system-wide notifications, and maintaining operational parameters. **Comprehensive reporting** provides administrators with detailed statistical analysis of borrowing rates, overdue materials, and user activity patterns across the entire system.

The system integrates **intelligent chatbot support** to provide automated assistance for frequently asked questions and system navigation guidance, enhancing user experience and reducing staff workload.

3.1.2 Non-Functional Requirements

The system incorporates advanced technological capabilities that enhance operational efficiency and user experience through intelligent automation and multilingual support.

Automatic translation services provide comprehensive language support by translating book information and user interface elements across multiple languages, ensuring accessibility for diverse user populations. **Quality assessment functionality** analyzes user reviews and feedback data to determine content quality ratings and inform collection development decisions.

Intelligent query processing enables the system to respond to complex questions regarding books, authors, and content through natural language processing capabilities. **Smart search assistance** supports natural language queries and contextual understanding, allowing users to locate materials using conversational search terms rather than rigid keyword structures.

Automated notification systems manage communication workflows by sending email reminders for overdue materials, ensuring timely returns and reducing administrative burden. **Statistical reporting automation** generates comprehensive reports analyzing borrowing trends, inventory status, and usage patterns without manual intervention.

Barcode and QR code management streamlines material handling through automated scanning and processing capabilities for efficient check-in and check-out procedures. **Image management systems** handle book cover uploads and organization, maintaining visual catalog presentation standards.

Multilingual interface support enables users to switch between different language interfaces according to their preferences, promoting inclusive access to library services. **Email notification integration** ensures users receive timely communications regarding due dates, overdue materials, and system updates, maintaining effective communication channels between the library and its patrons.

These non-functional requirements collectively enhance system usability, operational efficiency, and user satisfaction while supporting the library's mission to provide accessible and comprehensive information services.

3.1.3 Use Case Analysis

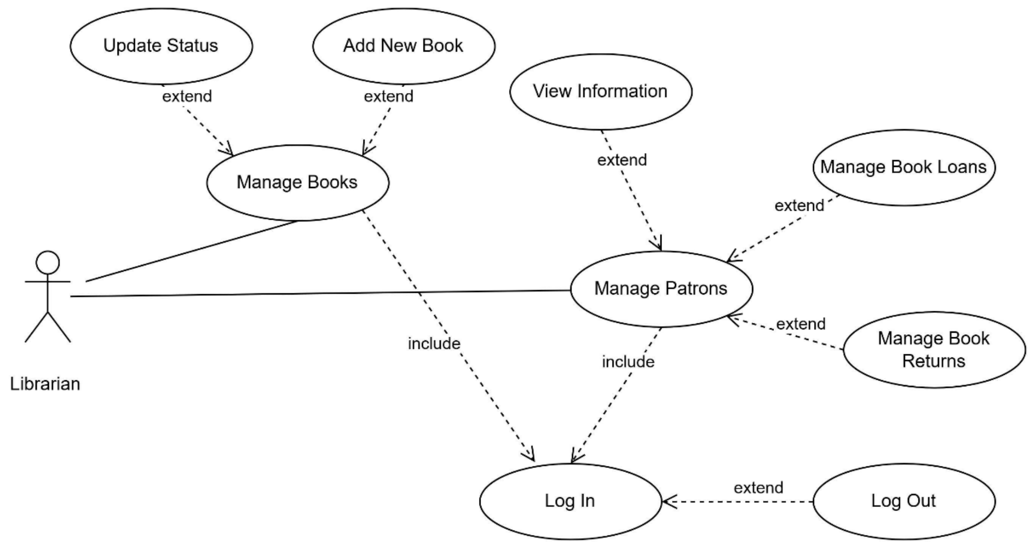


Figure 1 Usecase _ Librarian

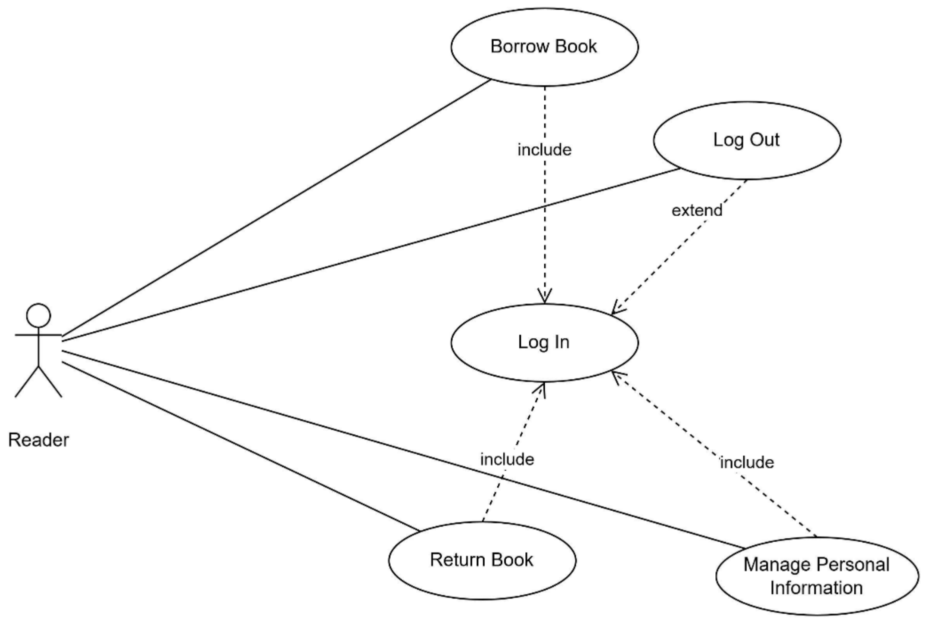


Figure 2 Usecase_Admin

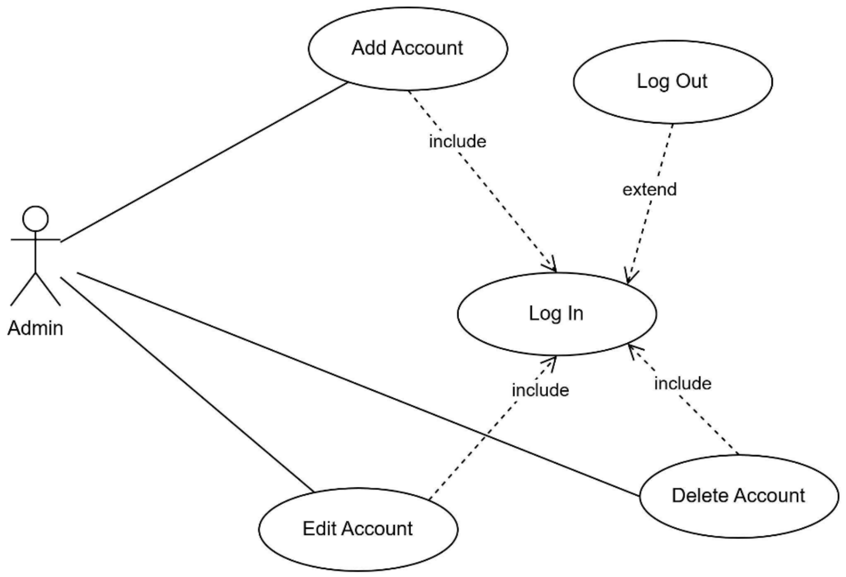


Figure 3 Usecase_Reader

STT	Use Case	Description
1	**Borrow book**	The Login function authenticates users by validating their credentials
2	**Manage Personal Information**	The Manage Personal Information feature allows users to view, edit, and update their personal details
3	**Manage Book**	The Manage Book feature enables users (librarian) to add, view, edit, delete, and search for book details in the system
4	**Manage Patron**	The Manage Patron feature enables librarians or administrators to add, view, edit, delete, and search library patron information
5	**Log in**	The Login feature allows users (members, administrators, librarians) to access the system by authenticating their accounts via username/email and password.

Figure 4 Main Functions

ID and Name:	UC-1 Login		
Created By:	Library System Team		
Primary Actor:	User		
Description:	A User accesses the Library Management System desktop application, enters their credentials, and logs into the system to access library services and manage their borrowing activities.		
Trigger:	User launches application and initiates login		
Preconditions:	PRE-1: User has a registered account in the database PRE-2: User has valid credentials		
Postconditions:	POST-1: User is authenticated and logged POST-2: System displays the appropriate user interface based on User permissions POST-3: User's session is established and tracked by the system		
Normal Flow:	1.0 Login to System 1. User launches the desktop application 2. Web displays the login form requesting username and password 3. User enters their username and password 4. User submits the login credentials 5. Web validates the credentials against the database 6. Web displays the User's personalized dashboard with their current borrowings, reserved books, and notifications 7. Web records the login time and IP address for security tracking in the database		
Alternative Flows:	1.1 Login with Library Card 1. User selects "Login with Library Card" option 2. User connects card reader and scans Library Card 3. Return to step 4 of normal flow 1.2 Reset Password 1. User selects "Forgot Password" option 2. Web guides User through password reset process 3. Return to step 1 of normal flow		
Exceptions:	1.0.E1 Invalid credentials 1. Web informs User that the username or password is incorrect		

	2a. If User cancels login process, then Web closes the login form 2b. Else if User attempts login again, Web allows retry 1.0.E2 Account locked 1. Web informs User that their account is locked due to multiple failed login attempts 2a. If User cancels login process, then Web terminates use case 2b. Else if User requests account unlock, Web provides instructions
Priority:	High
Frequency of Use:	Approximately 50 users, average of three usages per day. Peak usage load for this use case is between 3:00 P.M. and 7:00 P.M. local time.
Business Rules:	1.Account must exist in the system 2.Account must be active (not locked/disabled)
Other Information:	1. User shall be able to cancel the login process at any time prior to completion 2. The system should maintain a secure connection throughout the login process and subsequent session 3. After 15 minutes of inactivity, the system will automatically log the User out
Assumptions:	Assume that 80 percent of Users will use the username/password method rather than library card number (source: previous 6 months of system data).

Figure 5 UC1-Login

ID and Name:	UC-2 Borrow Book		
Created By:	Library System Team		
Primary Actor:	User		
Description:	A User accesses the library management system, searches for and selects a book to borrow, then completes the borrowing process to take the book.		
Trigger:	User has logged in and selected a book to borrow.		
Preconditions:	PRE-1: The User is logged into the system. PRE-2: The User has no overdue books or unpaid fines. PRE-3: The User has not reached their maximum borrowing limit. PRE-4: The desired book has an "Available" status.		
Postconditions:	POST-1: The book's status is marked as "On Loan" in the system.		

	POST-2: A new borrowing record is created in the database, including User ID, Book ID, borrowing date, and due date. POST-3: The system updates the User's total borrowed books count. POST-4: The system displays a confirmation message for successful borrowing and the due date.
Normal Flow:	1. The User is logged into the system. 2. The system displays the User's main interface. 3. The User searches for a book by entering a title, author, or book ID into the search bar. 4. The system displays the search results. 5. The User selects the desired book from the results list. 6. The system displays the book's details, including its current status (e.g., "Available," "On Loan," "Reserved"). 7. The User clicks the "Borrow Book" button. 8. The system checks the book's status. 9. The system automatically prints a borrowing receipt
Alternative Flows:	8.1 If the book's status is "On Loan" or "Reserved" the system informs the User: "Book is not available for borrowing. 8.2 The User can choose to reserve the book or return to select a different book.
Exceptions:	2.0.E1 No search results found 1. LMS informs User that no books match the specified criteria 2a. If User cancels search process, then LMS terminates use case 2b. Else if User modifies search terms, LMS executes new search
Priority:	High
Frequency of Use:	Frequent, depending on library activity.
Business Rules:	BR-1: Each user has a maximum number of books they can borrow concurrently. This limit may vary based on membership type (e.g., standard, premium). BR-2: A book can only be borrowed if its status is "Available."
Other Information:	- The system should maintain a secure connection throughout the borrowing process. - After 15 minutes of inactivity, the system will automatically log the User out.

Figure 6 UC2_ Borrow Books

ID and Name:	UC-3 Manage Personal Information		
Created By:	Library System Team		
Primary Actor:	User		
Description:	- The User accesses their personal information in the library management system, views their details, and can update certain information as needed.		
Trigger:	The User selects the "Manage Personal Information" option after logging in.		
Preconditions:	PRE-1: The User is logged into the system. PRE-2: The User has valid permissions to view/modify their own information.		
Postconditions:	POST-1: The system displays the User's current personal information. POST-2: If updates are made, the system saves the changes to the database. POST-3: The system confirms successful updates (if applicable).		
Normal Flow:	1. The User logs into the system. 2. The User navigates to the "My Account" or "Profile" section. 3. The system retrieves and displays the User's personal information (e.g., name, contact details, membership type, borrowing history). 4. The User selects "Edit Information" (if updates are needed). 5. The system allows modifications to editable fields (e.g., phone number, email, password). 6. The User confirms the changes. 7. The system validates the updated information. 8. The system saves the changes to the database. 9. The system displays a confirmation message: "Your information has been updated successfully."		
Alternative Flows:	5.1 If the User cancels the edit process, the system returns to the view mode without saving changes. 7.1 If validation fails (e.g., invalid email format), the system prompts the User to correct the input and resubmit.		
Exceptions:	EX-1: Database connection error → System displays: "Unable to save changes. Please try again later." EX-2: Unauthorized modification attempt (e.g., User tries to change restricted fields like User ID) → System logs the attempt and restricts access.		
Priority:	Medium		

Frequency of Use:	Occasional (as needed by the User).
Business Rules:	BR-1: Users can only modify their own information, not other users' data. BR-2: Certain fields (e.g., membership type, User ID) are non-editable by the User and require admin assistance. BR-3: Password changes require re-authentication for security.
Other Information:	- The system must encrypt sensitive data (e.g., passwords) before storage. - Changes to contact information (e.g., email) may require verification via confirmation link. - Session timeout after 15 minutes of inactivity applies.

Figure 7 UC-3 Manage Personal Information

ID and Name:	UC-4 Manage Book Catalog		
Created By:	Library System Team		
Primary Actor:	Staff		
Description:	The staff adds, edits, deletes, or searches for books in the library catalog, ensuring book information is accurate and up-to-date.		
Trigger:	The staff logs into the system and selects the "Manage Book Catalog" function.		
Preconditions:	PRE-1: The staff is logged in with appropriate privileges. PRE-2: The book catalog exists and is modifiable.		
Postconditions:	POST-1: New book information is added to the catalog POST-2: Existing book information is updated or removed POST-3: The system displays a success confirmation or error message		
Normal Flow:	1. The staff accesses the "Manage Catalog" section. 2. The system displays the current list of books (with pagination/search options). 3. The staff performs one of the following actions: 3.1 Add a New Book: 3.2 Edit a Book: 3.3 Search/Filter Books: 3.4 Search/Filter Books: 4. The system displays a success message ("Book added/updated/deleted successfully!").		

Alternative Flows:	3.1.1 If the book's ISBN already exists: - The system alerts: "This book already exists! Update the quantity instead of adding a new entry." 3.3.1 If the book is currently borrowed: - The system prevents deletion and displays: "Cannot delete a book that is on loan!" 3.4.1 If no books match the search: - Displays: "No results found."
Exceptions:	EX-1: Database connection error → Displays: "Connection error. Please try again later." EX-2: Unauthorized access → Blocks the action.
Priority:	Medium
Frequency of Use:	Occasional (as needed by the Staff).
Business Rules:	BR-1: Each book must have a unique ISBN. BR-2: A book cannot be deleted if: - It is currently borrowed (status = "On Loan"). - It has pending reservations (status = "Reserved"). BR-3: Book genres must be from a predefined list (e.g., Science, Literature).
Other Information:	- Security: Only Admins can modify the catalog. - Audit Logs: Records changes (who/when). - Frequency of Use: High (when new books arrive or updates are needed).

Figure 8 UC-4 Manage Book Catalog

3.2 System Design

3.2.1 System Architecture Overview

Our Library Management System is designed based on a modern 3-tier architecture, which ensures a clear separation between the user interface, business logic, and data storage.

The Presentation Layer is an Intelij Application responsible for rendering the user interface and handling user interactions. The Application Layer consists of a back-end server that acts as the core brain of the system; it processes all business logic requests, such as book searches, user authentication, and the management of check-out and return processes. Finally, the Data Layer comprises the relational database management system, which is responsible for the persistent and secure storage and retrieval of all library data. This architecture not only facilitates easier maintenance and scalability but also allows each tier to be developed and replaced independently of the others.

3.2.2 Class Diagram

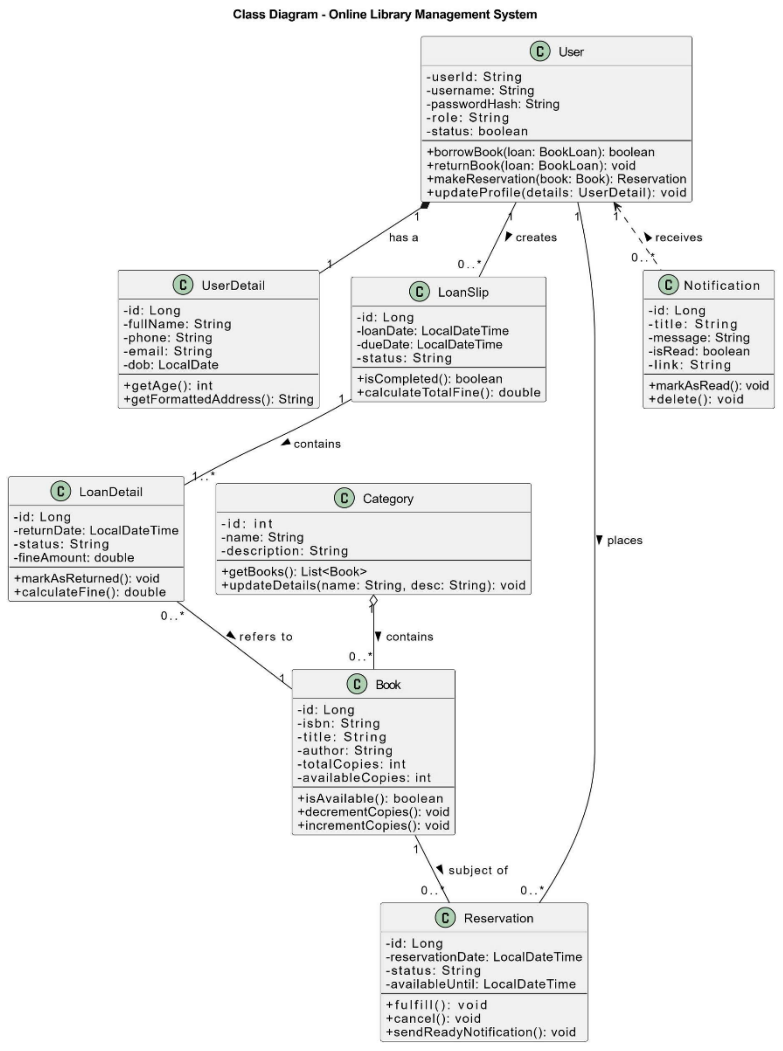


Figure 9 Class Diagram - Online Library Management System

Main Object Classes:

+ **User:** The central class representing system users. This class manages login credentials (username, passwordHash), roles, and key actions such as borrowing books (borrowBook), returning books (returnBook), and making reservations (makeReservation).

+ **Book:** Represents a book in the library, containing basic information such as title, author, ISBN, and—most importantly—totalCopies (total inventory) and availableCopies (copies currently available) for stock management.

+ **LoanSlip:** Acts as the master object for a loan transaction. It contains general information such as the borrower (`User`), the loan date, and the due date.

+**LoanDetail:** Is the detail object containing information for each specific book (`Book`) within a loan slip. It manages the return status and fine for that individual book.

+ **Reservation:** Represents a user's request to reserve a book when it is out of stock.

+ **Category:** Used to classify books by genre, facilitating easier management and search.

+ **UserDetail:** Stores personal details of users, separated from account information to enhance security and clarity.

+ **Notification:** Represents system notifications sent to users (e.g., return reminders, reservation availability alerts).

Key Relationships:

+ **User and UserDetail (1-to-1):** A "has-a" relationship. One User has one UserDetail. This separation is a sound design choice, clearly distinguishing authentication data from personal information.

+ **User and LoanSlip / Reservation (1-to-Many):** A User can "create" multiple LoanSlip records and "place" multiple Reservation requests. This accurately reflects core user actions.

+ **Book and Reservation (1-to-Many):** A Book can be "part of" multiple loans and the "subject of" multiple reservations.

+ **Book and LoanDetail (1-to-Many):** A single Book can be referred to by many (0..*) LoanDetail records over time, reflecting its borrowing history. Each LoanDetail, however, refers to exactly one Book.

+ LoanSlip and LoanDetail (1-to-Many): This is the central relationship in the new design. A LoanSlip contains (contains) one or more (1..*) LoanDetails. This ensures that a loan slip cannot exist without at least one book being borrowed.

+ **Category and Book (1-to-Many): A** "contains" relationship. One Category can contain multiple Book entries, enabling organized cataloging.

+ **User and Notification (1-to-Many):** A User can be the "recipient" of multiple Notification alerts.

3.2.3 Sequence Diagrams

1. User Registration Flow

This diagram illustrates the process of a new user registering an account on the system. The flow begins when the user submits information from the registration form. The Web System receives the request and performs a username check against the Database. Based on the result, the system handles two scenarios: if the username already exists, an error message is returned; if it is valid, the user's information is saved, and a successful registration message is displayed.

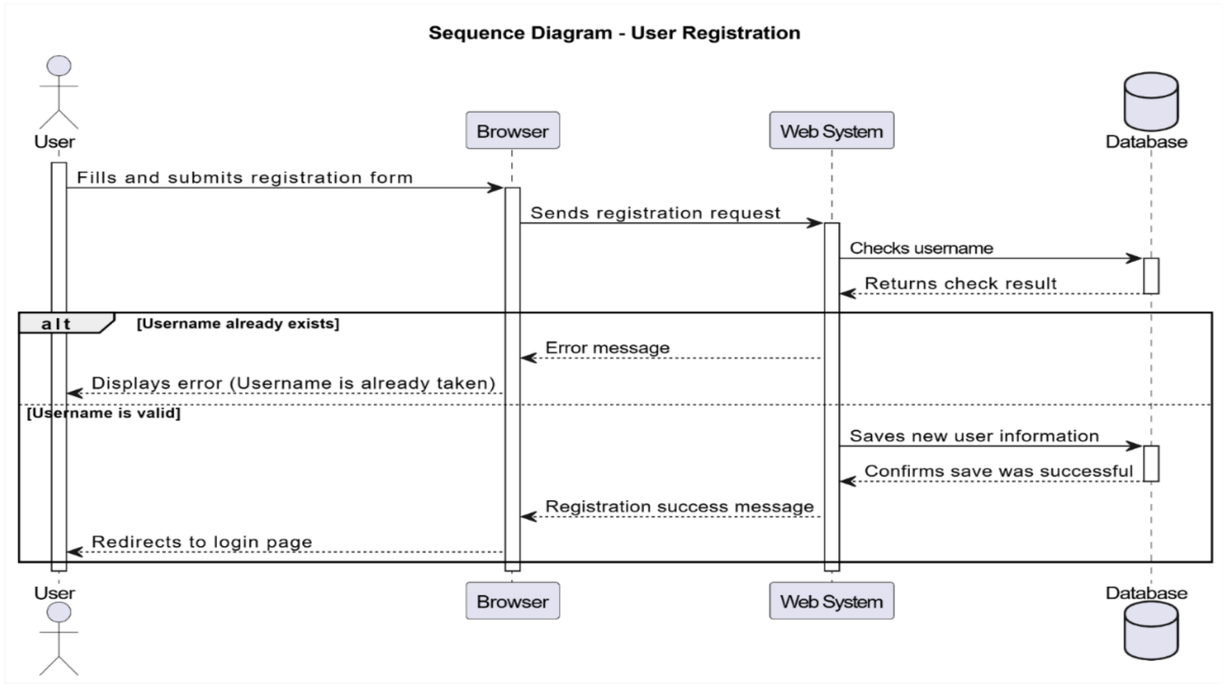

Sequence Diagram - User Registration
User
Browser
Web System
Database
Fills and submits registration form
Sends registration request
Checks username
Returns check result
alt
[Username already exists]
Error message
Displays error (Username is already taken)
[Username is valid]
Saves new user information
Confirms save was successful
Registration success message
Redirects to login page

Figure 10 Sequence Diagram - User Registration

2. Book Borrowing Flow

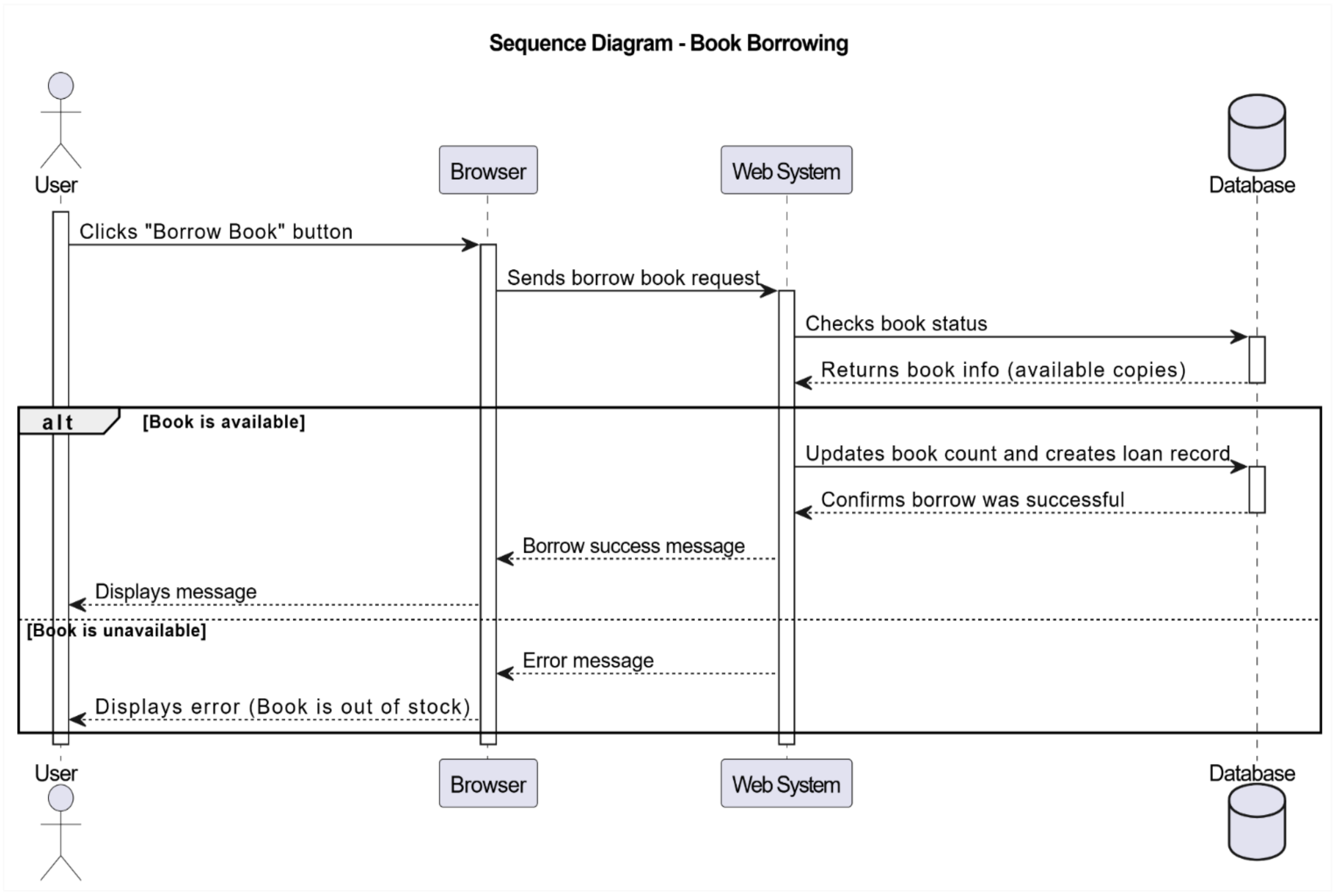

Sequence Diagram - Book Borrowing
User
Browser
Web System
Database
Clicks "Borrow Book" button
Sends borrow book request
Checks book status
Returns book info (available copies)
alt
[Book is available]
Updates book count and creates loan record
Confirms borrow was successful
Borrow success message
Displays message
[Book is unavailable]
Error message
Displays error (Book is out of stock)

Figure 11 Sequence Diagram - Book Borrowing

This diagram describes the core business process when a member (Reader) requests to borrow a book. When the user clicks the "Borrow" button, the Web System checks the book's status in the Database. If the book is available, the system updates the quantity, creates a new loan record, and notifies the user. Conversely, if the book is out of stock, the user will receive an error message.

3. Librarian Adds a New Book Flow

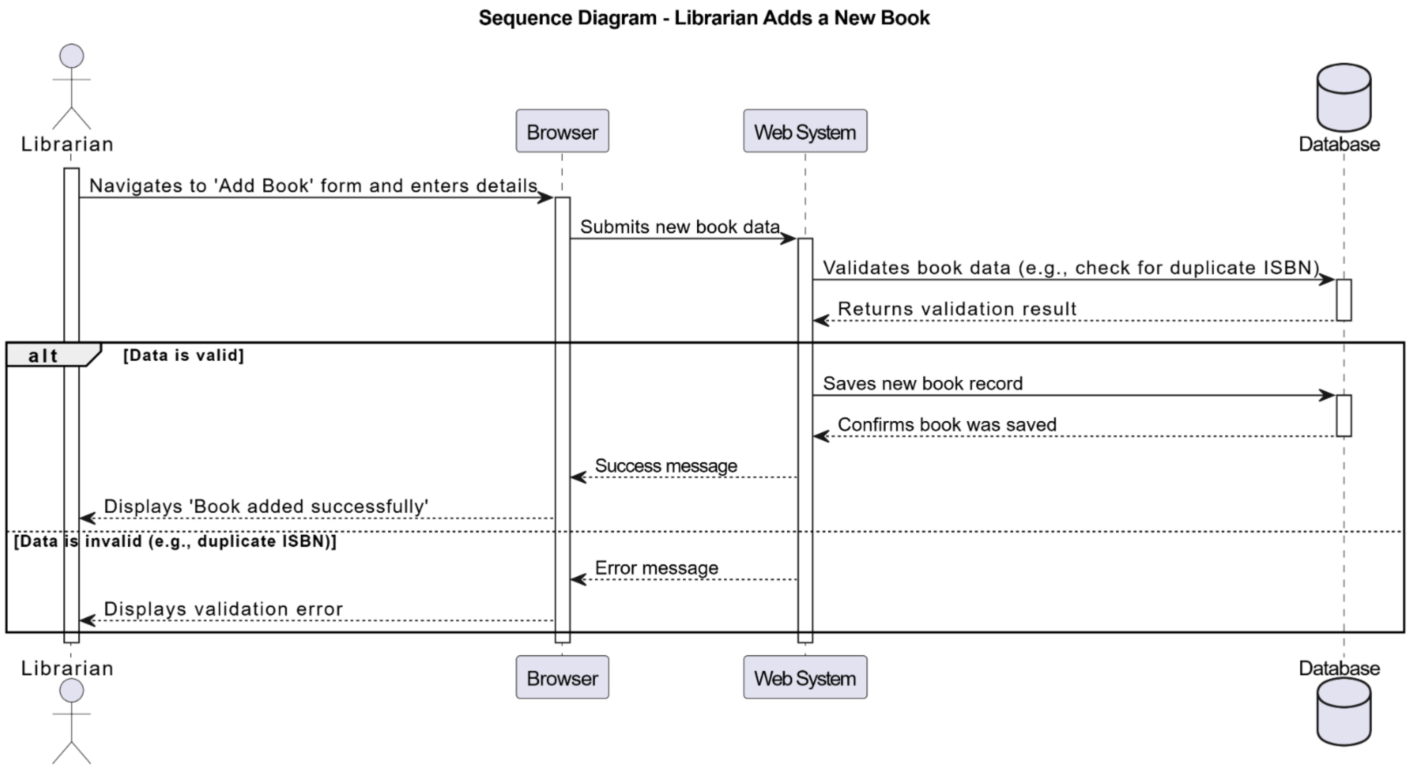

Sequence Diagram - Librarian Adds a New Book
Librarian
Browser
Web System
Database
Navigates to 'Add Book' form and enters details
Submits new book data
Validates book data (e.g., check for duplicate ISBN)
Returns validation result
alt
[Data is valid]
Saves new book record
Confirms book was saved
Success message
Displays 'Book added successfully'
[Data is invalid (e.g., duplicate ISBN)]
Error message
Displays validation error

Figure 12 Sequence Diagram - Librarian Adds a New Book

This diagram shows the workflow for a librarian or staff member adding a new book to the library's catalog. After logging in, the librarian fills out a form with the new book's details (ISBN, title, author, etc.) and submits it. The Web System receives this information, validates it (e.g., checking if the ISBN is unique), and if the data is valid, saves the new book record to the Database. A success message is then returned to the librarian.

3.2.4 Activity Diagrams

The **"Manage Personal Information**" function within the system allows users to log in and view their current personal details. Users have the option to edit this information. After making changes, the system will validate and check the entered data. If any errors are found, an error message will be displayed, prompting the user to correct the information. Otherwise, if the data is valid, the information will be successfully updated, and a confirmation message will be shown.

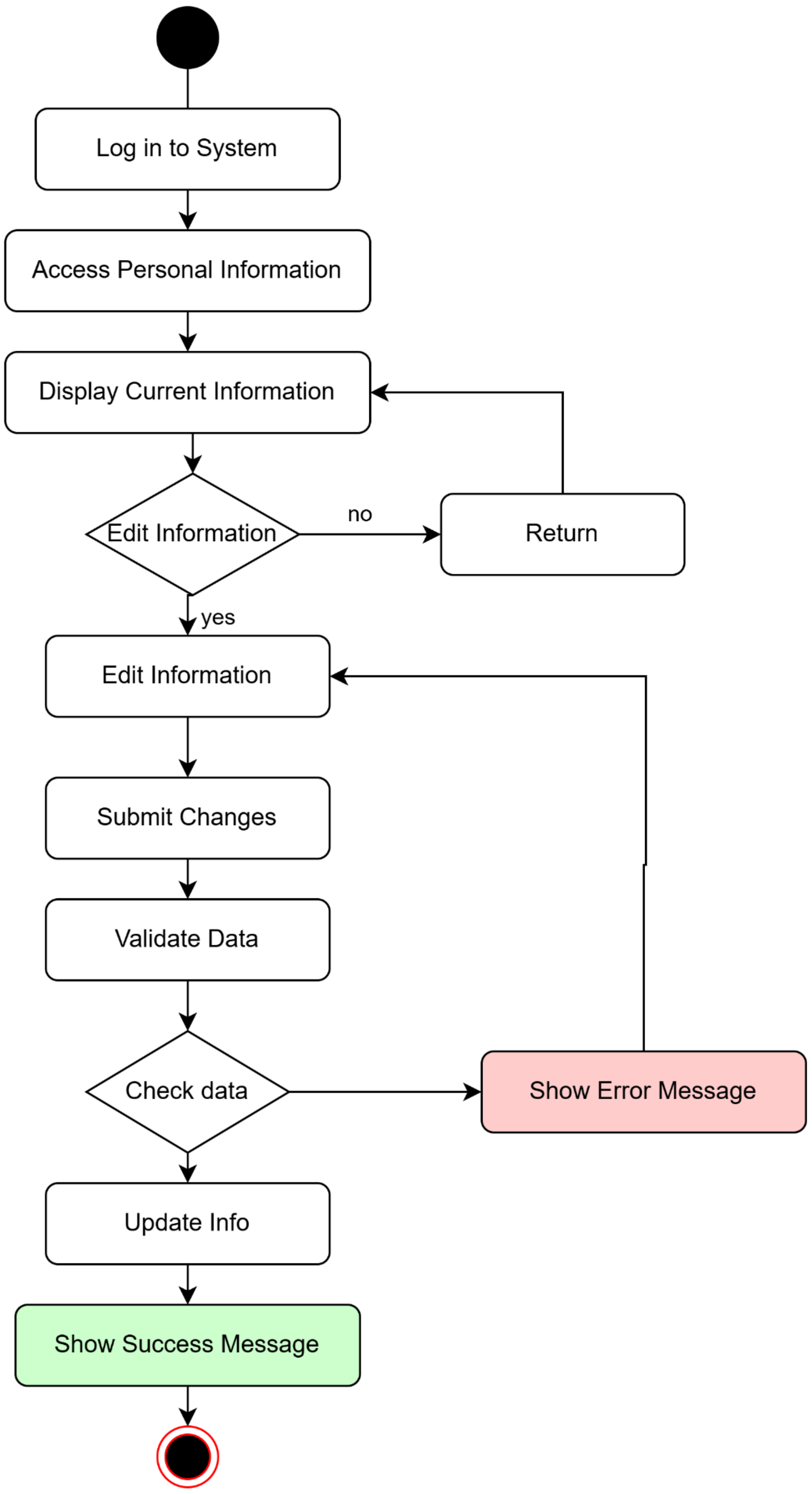


Figure 13 Manage Personal Information

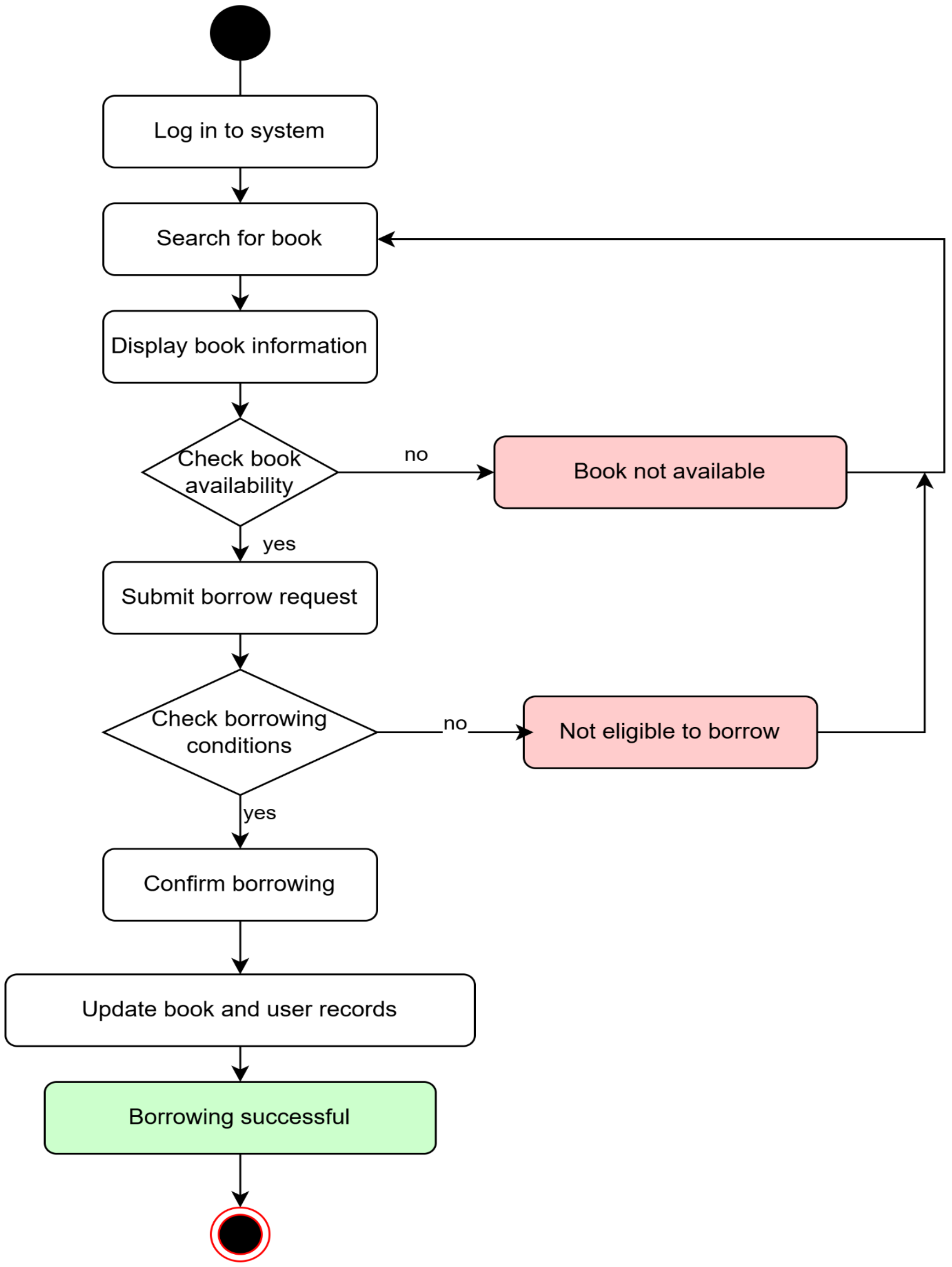


Figure 14 Borrow books

The book borrowing process begins when the user logs in and searches for a book. The system will check the book's availability and the user's eligibility to borrow. If both conditions are valid, the user confirms the request, the system updates the records, and then notifies that the borrowing was successful. Conversely, if the book is unavailable or the user is not eligible, the process prompts the user to go back and search for a different book.

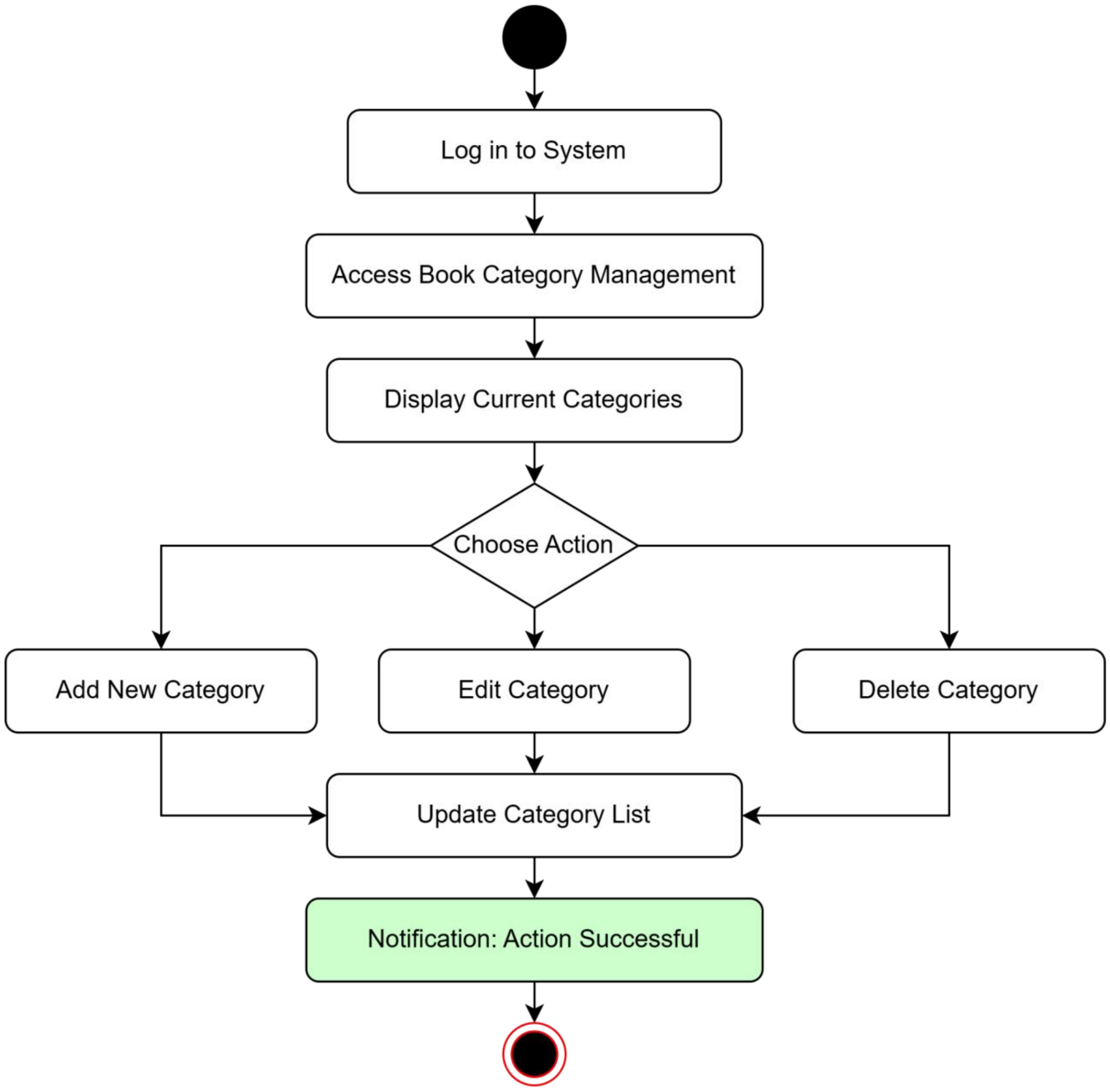


Figure 15 Manage Book Categories

The **"Manage Book Categories"** function in the system allows logged-in users to access and view the list of existing categories. Users can perform management operations such as adding new categories, editing existing category information, or deleting categories. After any changes are made, the category list will be updated, and a successful action notification will be displayed.

3.2.5 State Diagram

This state diagram illustrates the lifecycle of a book within a library management system, focusing on its availability. Initially, a book is in the "Available" state. It transitions to the "Fully Loaned" state when the last copy is borrowed. If a reserved user comes to borrow the book, it moves from "Fully Loaned" to "Reserved." A book returns to the "Available" state when it's returned without any prior reservations, or when a reservation is canceled or expires. This diagram ensures the system accurately reflects the actual status of each book.

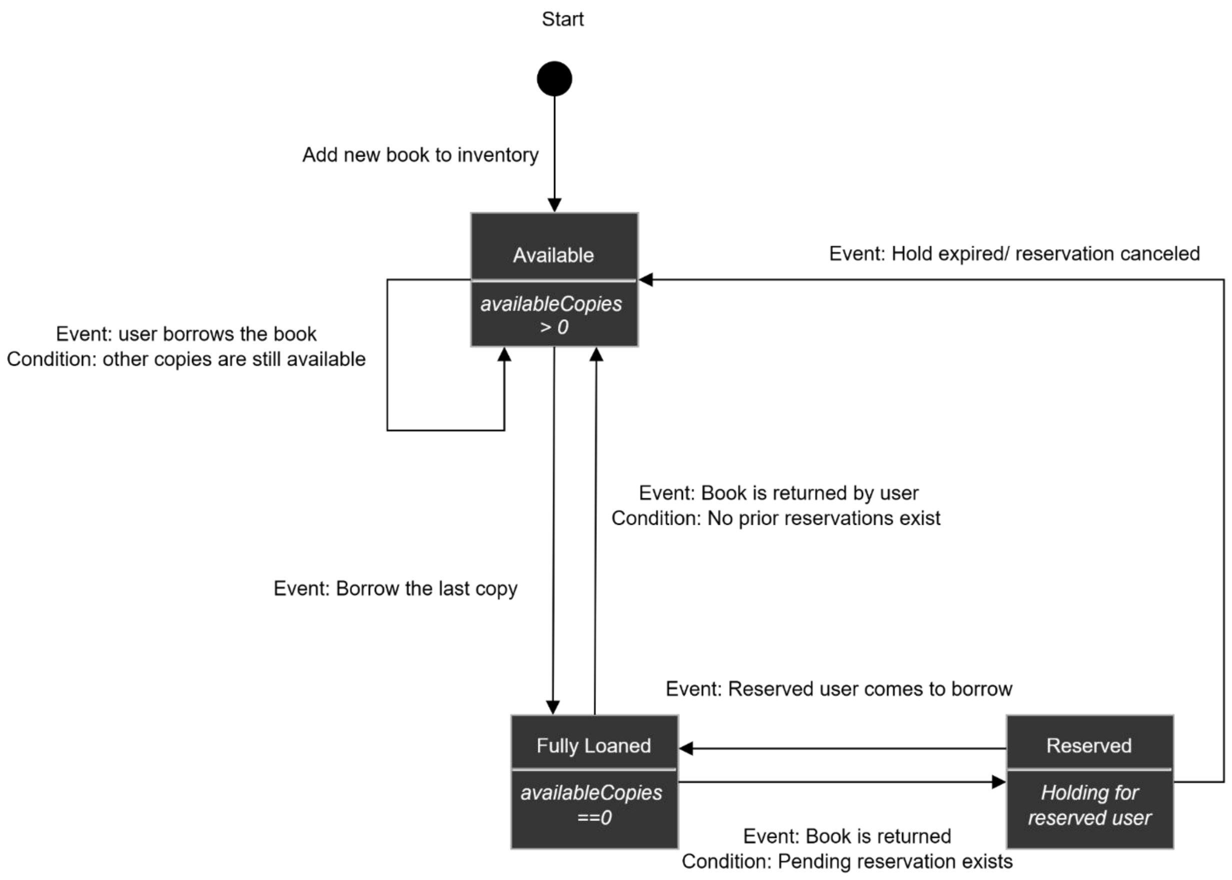

Start
Add new book to inventory
Available
availableCopies > 0
Event: Hold expired/ reservation canceled
Event: user borrows the book
Condition: other copies are still available
Event: Book is returned by user
Condition: No prior reservations exist
Event: Borrow the last copy
Event: Reserved user comes to borrow
Fully Loaned
availableCopies ==0
Reserved
Holding for reserved user
Event: Book is returned
Condition: Pending reservation exists

Figure 16 State Diagram_1

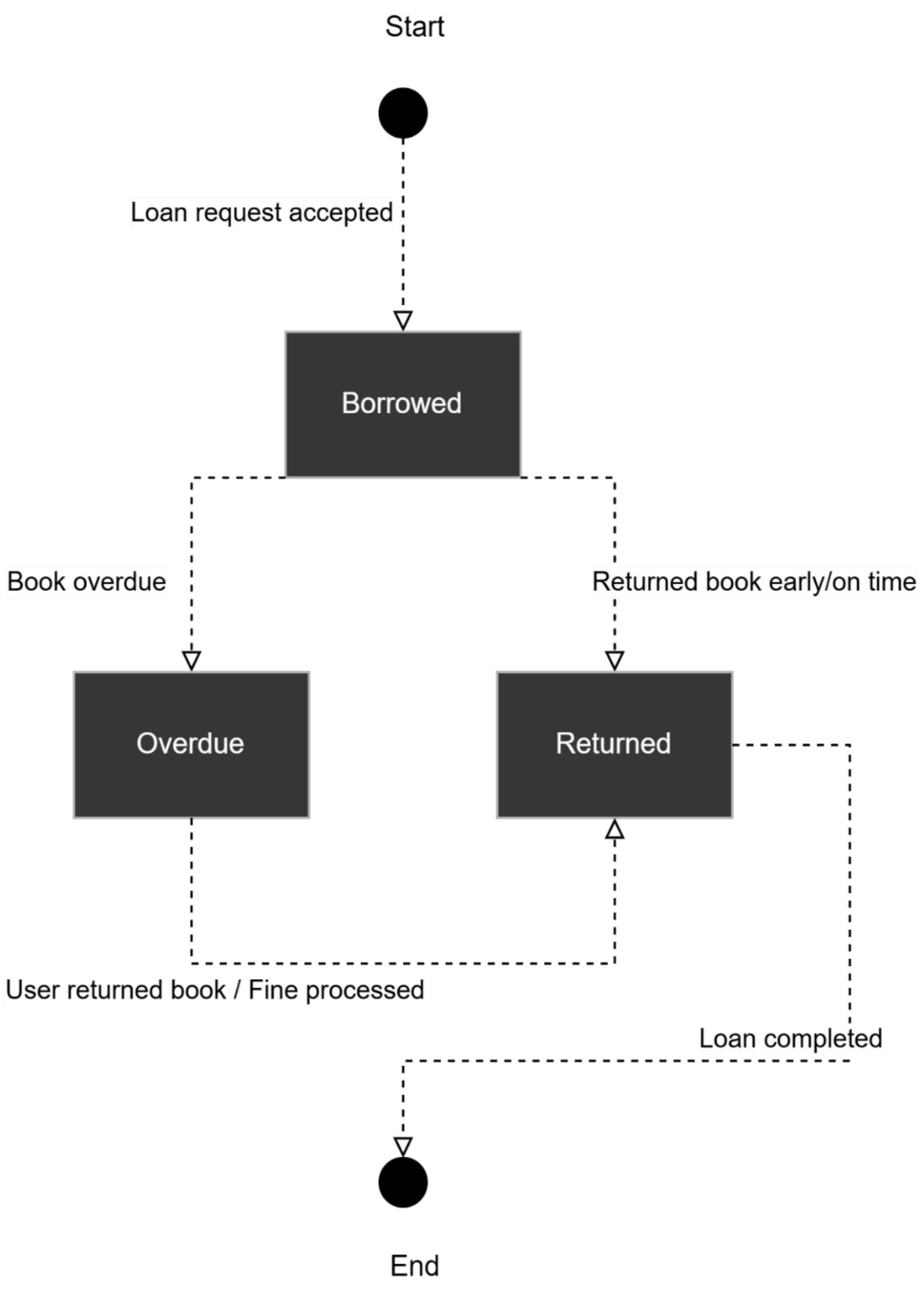


Figure 17 State Diagram_2

This state diagram illustrates the main stages of a book after a loan request is accepted within a library system. Initially, when a loan request is accepted, the book transitions to the **"Borrowed"** state. From this state, there are two possibilities: if the book is returned early or on time, it moves to the **"Returned"** state. However, if the book becomes overdue, it will transition to the "Overdue" state. When the book is in the **"Overdue"** state and the user returns the book or the fine is processed, the book will also move back to the "Returned" state. Finally, once the loan is completed, the book's lifecycle ends. This diagram clearly illustrates the process of tracking a book's status from when it's borrowed until the return process is finalized.

3.3 Database Design

- ### 3.3.1 Database Schema

The database schema is built upon the entities and relationships identified during the analysis phase. The tables are grouped into major logical clusters: user management, book management, and transaction management (borrowing, returning, reserving). This design ensures that data is normalized, redundancy is minimized, and consistency is maintained.

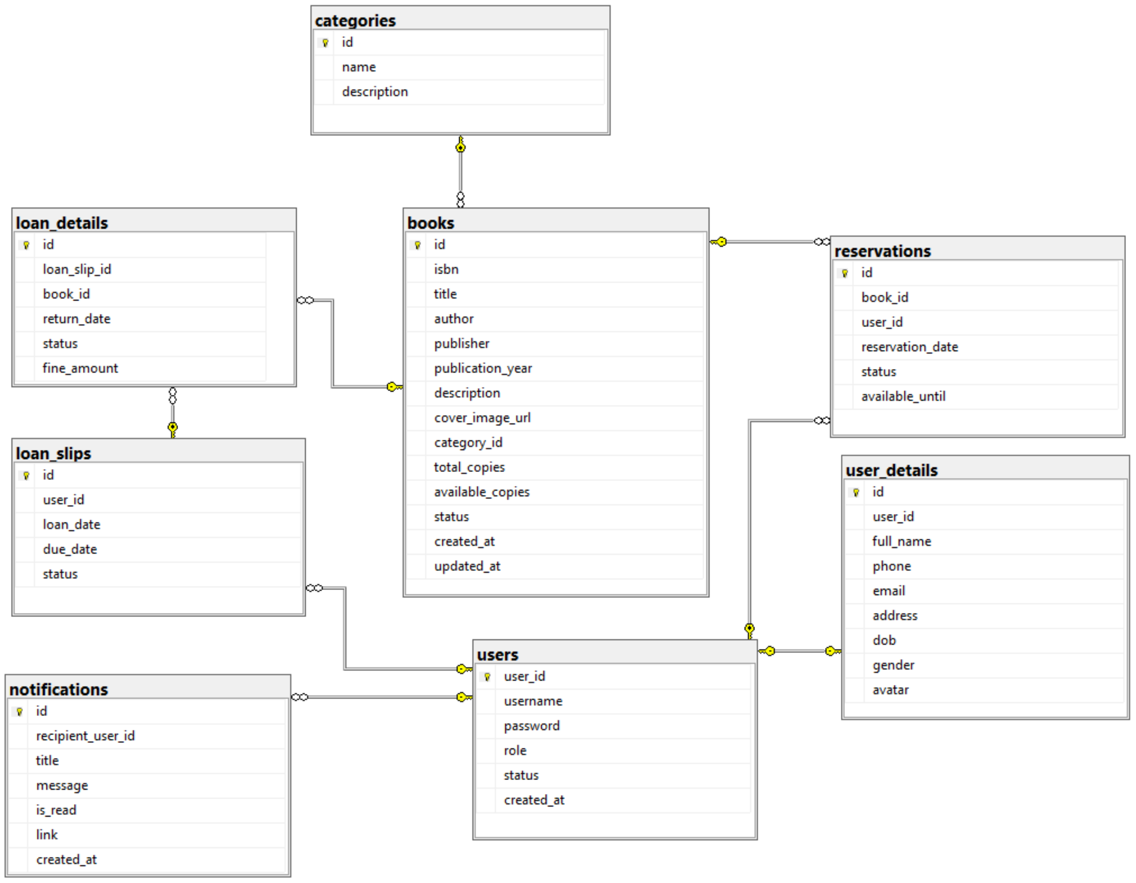


Figure 18 Database

3.3.2 Table Structures

1. Users table

This is the central user management table, storing the necessary account information for authentication and authorization within the system.

Atrribute Name	Description	Data Type	Key/Constraint
user_id	The unique identifier for the user.	NVARCHAR(255)	**Primary Key (PK)**, NOT NULL
username	The user's login name.	NVARCHAR(50)	UNIQUE, NOT NULL
password	The user's encrypted password.	NVARCHAR(255)	NOT NULL
role	The user's role (READER, STAFF, ADMIN).	NVARCHAR(20)	NOT NULL, CHECK IN ('ADMIN', 'STAFF', 'READER')
status	The account status (1: Active, 0: Inactive).	BIT	NOT NULL, DEFAULT 1
created_at	The timestamp when the account was created.	DATETIME2	NOT NULL, DEFAULT GETDATE()

Figure 19 User Table

2. User_details table

This table stores the detailed personal information of users, separated from the Users table for optimization and security.

Atrribute Name	Description	Data Type	Key/Constraint
id	The auto-incrementing identifier for the table.	BIGINT	**Primary Key (PK)**, IDENTITY
user_id	The user ID, linking to the users table.	NVARCHAR(255)	Foreign Key (FK), UNIQUE, NOT NULL
full_name	The user's full name.	NVARCHAR(255)	NULL
phone	Phone number.	NVARCHAR(20)	NULL
email	Email address.	NVARCHAR(255)	UNIQUE, NULL
address	Address.	NVARCHAR(255)	NULL
dob	Date of birth.	DATE	NULL
gender	Gender of the user.	NVARCHAR(10)	CHECK (gender IN ('Male', 'Female', 'Other')), DEFAULT 'Other'

Figure 20 User_details table

3. Books Table

Stores all information about the books available in the library.

Atrribute Name	Description	Data Type	Key/Constraint
id	The auto-incrementing identifier for the book.	BIGINT	**Primary Key (PK)**, IDENTITY
isbn	The International Standard Book Number.	NVARCHAR(20)	UNIQUE, NOT NULL
title	The title of the book.	NVARCHAR(255)	NOT NULL
author	The author of the book.	NVARCHAR(255)	NOT NULL
publisher	The publisher.	NVARCHAR(255)	NULL
publication_year	The year of publication.	INT	NULL
description	A description of the book's content.	NVARCHAR(MAX)	NULL
category_id	The category ID, linking to the **categories table.**	INT	**Foreign Key (FK),** NULL
total_copies	The total number of copies for this book.	INT	NOT NULL, DEFAULT 1

available_copies	The number of copies currently available to borrow.	INT	NOT NULL, DEFAULT 1
status	The condition of the book (AVAILABLE, DAMAGED, LOST).	NVARCHAR(50)	DEFAULT 'AVAILABLE'
created_at	The timestamp when the book was added.	DATETIME2	NOT NULL, DEFAULT GETDATE()
updated_at	The timestamp when the book info was last updated.	DATETIME2	NOT NULL, DEFAULT GETDATE()

Figure 21 Books Table

4. Categories Table

Used to classify books into different categories (e.g., Science, Literature).

Atrribute Name	**Description**	**Data Type**	**Key/Constraint**
id	The auto-incrementing identifier for the category.	INT	Primary Key (PK), IDENTITY
name	The name of the category.	NVARCHAR(100)	UNIQUE, NOT NULL
description	A detailed description of the category.	NVARCHAR(500)	NULL

Figure 22 Categories Table

5. Loan_slips Table

This table acts as the master table, storing general information for each book borrowing transaction. Each transaction creates a unique slip.

Atrribute Name	Description	Data Type	Key/Constraint
id	The auto-incrementing identifier for the loan.	BIGINT	Primary Key (PK), IDENTITY
user_id	The ID of the user performing the loan.	NVARCHAR(255)	Foreign Key (FK), NOT NULL
loan_date	The date the loan was made.	DATETIME2	NOT NULL, DEFAULT GETDATE()
due_date	The common due date for all books on the slip.	DATETIME2	NOT NULL
status	Slip status (ACTIVE, COMPLETED).	NVARCHAR(50)	NOT NULL

Figure 23 Loan_slips Table

6. Loan-details Table

The detail table that lists the specific books belonging to a loan slip.

Atrribute Name	Description	Data Type	Key/Constraint
id	The auto-incrementing identifier.	BIGINT	**Primary Key (PK),** IDENTITY

loan_slip_id	The loan slip ID, linking to ***loan_slips***.	BIGINT	Foreign Key (FK), NOT NULL
book_id	The ID of the book being reserved.	BIGINT	Foreign Key (FK), NOT NULL
return_date	The actual return date (NULL if not yet returned).	DATETIME2	NOT NULL
status	Status of each book (BORROWED, RETURNED).	NVARCHAR(50)	NOT NULL, DEFAULT 'BORROWED'
fine_amount	The fine amount (if any) for this book.	DECIMAL(10, 2)	DEFAULT 0.00

Figure 24 Loan_details Table

7. Reservation Table

Stores user requests to reserve a book when it is currently unavailable.

Atrribute Name	**Description**	**Data Type**	**Key/Constraint**
id	The auto-incrementing identifier for the reservation.	BIGINT	Primary Key (PK), IDENTITY
book_id	The ID of the book being reserved.	BIGINT	Foreign Key (FK), NOT NULL

user_id	The ID of the user making the reservation.	NVARCHAR(255)	Foreign Key (FK), NOT NULL
reservation_date	The date the reservation was made.	DATETIME2	NOT NULL, DEFAULT GETDATE()
status	The status (PENDING, AVAILABLE, FULFILLED, EXPIRED).	NVARCHAR(50)	NOT NULL, DEFAULT 'PENDING'
available_until	The deadline for the user to pick up the book.	DATETIME2	NULL

Figure 25 Reservation Table

8. Notification Table

Stores notifications sent to users (e.g., return reminders, reservation availability).

Atrribute Name	**Description**	**Data Type**	**Key/Constraint**
id	The auto-incrementing identifier for the notification.	BIGINT	Primary Key (PK), IDENTITY
recipient_user_id	The ID of the user receiving the notification.	NVARCHAR(255)	Foreign Key (FK), NOT NULL
title	The title of the notification.	NVARCHAR(255)	NOT NULL

message	The detailed content of the notification.	NVARCHAR(1000)	NOT NULL
is_read	The read status (1: Read, 0: Unread).	BIT	NOT NULL, DEFAULT 0
link	An optional link to click on from the notification.	NVARCHAR(255)	NULL
created_at	The timestamp when the notification was created.	DATETIME2	NOT NULL, DEFAULT GETDATE()

Figure 26 Notification Table

3.3.3 Relationships and Constraints

To ensure data integrity and consistency, the database defines the following relationships and constraints:

_One-to-One Relationship: users and **user_details**: Each record in the users table has exactly one corresponding record in the user_details table, linked via the user_id foreign key. ON DELETE CASCADE ensures that when a user is deleted, their details are also deleted.

_One-to-Many Relationships:

loan_slips and **loan_details:** One loan slip (loan_slips) can have many loan details (loan_details). This allows a single transaction to manage multiple books.

users and **reservations**: One user can make many book reservations.

users and **notifications**: One user can receive many notifications. ON DELETE CASCADE ensures all of a user's notifications are deleted when their account is removed.

categories and **books**: One category can contain many books. ON DELETE SET NULL ensures that if a category is deleted, books in that category are not deleted but their category_id is updated to NULL.

books and **loan_details/reservations**: One book title can be associated with many loans and reservations.

Data Integrity Constraints:

Primary Key (PK): Each table has a primary key to ensure every record is unique.

Foreign Key (FK): Used to establish links between tables and ensure referential integrity.

UNIQUE: Ensures that values in a column are unique, applied to fields like username, email, and isbn.

NOT NULL: Ensures that important fields are not left empty.

DEFAULT: Provides a default value for columns when no value is specified during data insertion.

CHECK: The gender column in user_details uses a CHECK constraint to limit values to 'Male', 'Female', or 'Other'.

3.4 System Implementation

3.4.1 Technology Stack

For our Library Management System, we have selected a modern and efficient technology stack to handle the specific needs of cataloging, member management, and circulation. The front-end is a single-page application built with Intelig , which provides a dynamic and responsive user interface for librarians and members to search the book catalog, view their borrowing history, and manage account details. All information is stored in a SQL Server relational database, chosen for its data integrity and ability to efficiently manage the relationships between books, members, and loan records.

3.4.2 Code Structure and Organization

The project's code is logically organized into modules based on the core domains of a library to ensure clarity and scalability. The primary directories are books and members, each encapsulating the specific logic for its domain. For example, the books module contains all the code related to catalog management, including search algorithms and validation, while the loans module handles all the logic for borrowing, returning, and calculating overdue fines. This modular, feature-driven approach allows developers to work independently on different parts of the system with minimal conflict. A shared core directory provides common functionalities like the database connection, authentication services to differentiate between member and librarian roles, and logging utilities. This structured organization, combined with consistent coding standards, makes the codebase easy to navigate, maintain, and extend with new features.

3.4.3 Key Features Implementation

The implementation of key features focused on creating a robust and user-friendly library experience. A central feature, the Book Circulation System, was implemented to manage the check-out and return process with full transactional integrity. When a book is borrowed, the system creates a new record in the loans table, linking a member_id to a book_id and setting a due_date. Critically, this operation atomically decrements the count of available copies in the books table to prevent conflicts. Another key feature is the Automated Overdue Notification System. This was implemented using a scheduled job that runs daily to scan the loans table for overdue items. Upon finding an overdue book, the system automatically triggers an email reminder to the corresponding member, significantly reducing the manual workload for librarians and ensuring timely returns.

3.5 User Interface Design

3.5.1 UI/UX Design Principles

The user interface design for the Library Management System is founded upon comprehensive user research and analysis of library personnel workflows, establishing design principles that prioritize operational efficiency while maintaining intuitive usability across diverse technological proficiency levels. The design methodology incorporates human-centered design principles that address the specific cognitive and physical requirements of library staff who interact with the system throughout their daily operational responsibilities.

User Research and Requirements Analysis

The design process initiated with extensive ethnographic research within academic library environments, conducting structured interviews with library administrators, circulation staff, and cataloging specialists to identify critical workflow patterns and usability requirements. This research revealed that library personnel require rapid task completion capabilities, minimal cognitive load during routine operations, and clear visual feedback for transaction confirmation. The analysis identified that users frequently multitask between patron assistance and system operations, necessitating interface designs that support task resumption and context preservation.

Design Principles and Visual Hierarchy

The visual design framework employs a clean, professional aesthetic that reduces visual clutter while maintaining comprehensive functionality access. The color palette utilizes a neutral base with strategic accent colors that provide clear visual distinction between different operational contexts. Primary actions receive prominent visual treatment through consistent button styling and placement, while secondary functions remain accessible through logical hierarchical organization.

Typography selection prioritizes readability across extended usage periods, employing sans-serif fonts with appropriate sizing and spacing that accommodate various lighting conditions typical in library environments. The information architecture implements progressive disclosure techniques that present essential information prominently while providing detailed data access through logical drill-down mechanisms.

User Experience Flow Design

The user experience design addresses common library workflows through streamlined task completion paths that minimize navigation complexity and reduce potential for operational errors. The system implements consistent interaction patterns across all functional modules, ensuring that users can apply learned behaviors from one system area to others without cognitive relearning requirements.

Error prevention strategies include real-time input validation, confirmation dialogs for irreversible actions, and clear labeling conventions that eliminate ambiguity in system functionality. The interface provides immediate feedback for all user actions through visual and textual confirmation mechanisms that maintain user confidence in system responsiveness.

The design incorporates WCAG 2.1 accessibility standards to ensure system usability across diverse user capabilities, including considerations for visual impairments, motor limitations, and cognitive processing differences. Color contrast ratios exceed recommended minimums while font sizing options accommodate various visual acuity requirements.

Keyboard navigation alternatives provide complete system functionality access without mouse dependency, supporting users with motor limitations or personal preference for keyboard-based interaction. The interface design maintains logical tab order and clear focus indicators that facilitate efficient keyboard navigation patterns.

3.5.2 Wireframes and Mockups

The wireframe development process employed systematic progression from low-fidelity conceptual layouts to high-fidelity interactive prototypes, ensuring comprehensive validation of design concepts before implementation commitment. This iterative approach facilitated early identification of usability issues while maintaining development timeline efficiency through structured design validation processes.

Initial wireframe creation focused on establishing optimal information architecture and navigation flow patterns without visual design distraction. The wireframes addressed critical screen layouts including the main dashboard, book catalog management, patron information systems, and transaction processing interfaces. Each wireframe underwent multiple iterations based on user feedback sessions that validated workflow alignment and identified potential usability challenges.

The wireframe development process utilized collaborative design sessions with library personnel to ensure that proposed layouts aligned with established operational procedures while introducing efficiency improvements through strategic interface optimization. These sessions revealed critical requirements for multi-window support and context switching capabilities that influenced final design decisions.

The transition from wireframes to detailed mockups incorporated comprehensive visual design elements while maintaining the validated information architecture established during wireframe development. The mockups demonstrate complete visual treatment including color schemes, typography implementation, iconography selection, and component styling that creates cohesive user experience across all system modules.

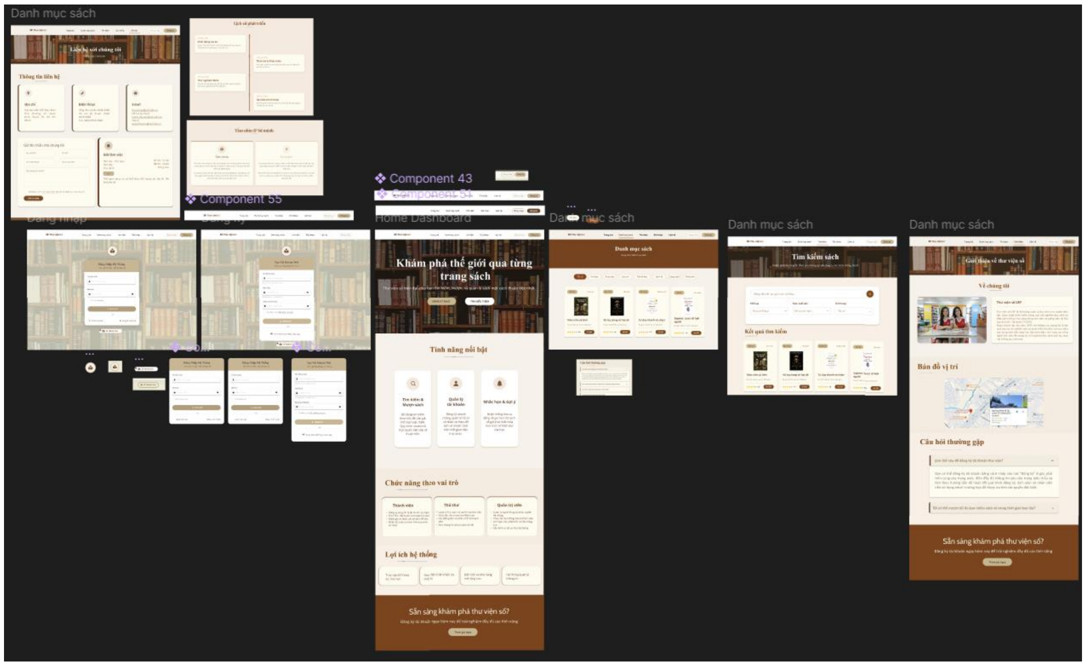
Interactive prototype development enabled comprehensive usability testing that validated design decisions under realistic usage scenarios. The prototypes incorporated realistic data sets and simulated system responses that provided accurate representation of final system behavior during user evaluation sessions.

Design Tool Implementation

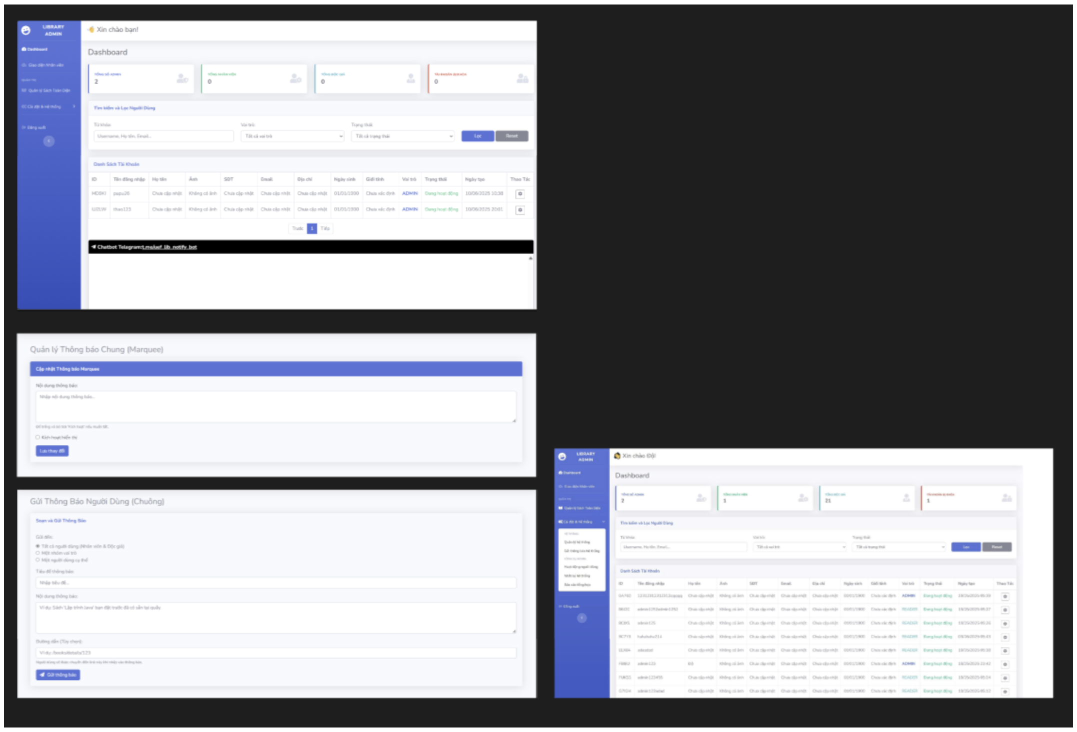
The design process utilized Figma as the primary design platform, enabling collaborative design development and comprehensive component library management that ensured consistency across all interface elements. The component-based design approach facilitated rapid iteration and maintained design system integrity throughout the development process.

Version control mechanisms within the design platform enabled systematic tracking of design evolution while maintaining access to previous iterations for comparative analysis and rollback capabilities when necessary. The collaborative features supported real-time feedback integration from stakeholders and development team members.

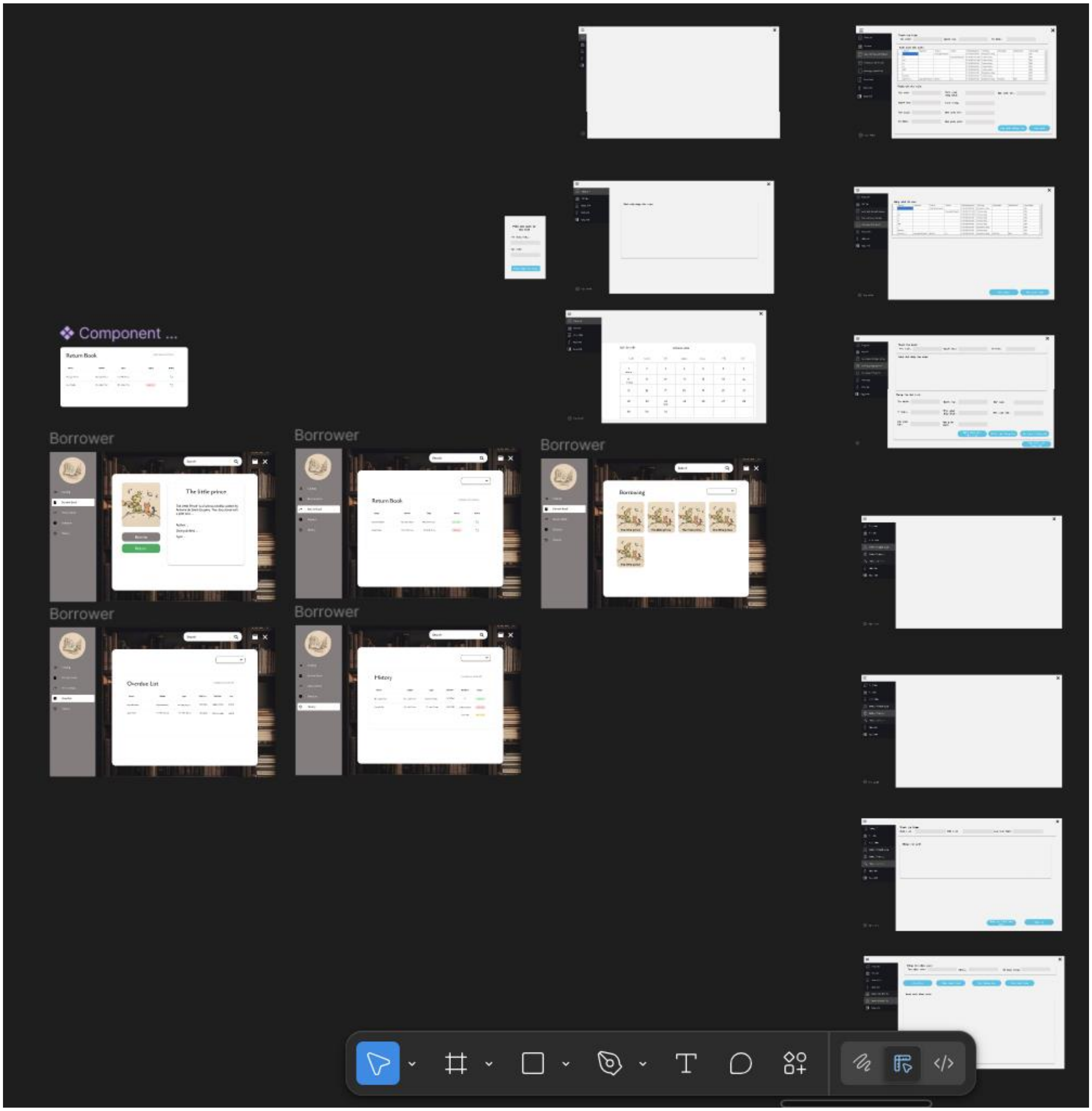
3.5.3 User Interface Implementation



Picture 1 Design Interface for Reader and Login/Register


LIBRARY ADMIN
Xin chào bạn!
Dashboard
Quản lý Thông báo Chung (Marquee)
Cập nhật Thông báo Marquee
Gửi Thông Báo Người Dùng (Chuông)
Gửi thông báo

Picture 2 Design Interface for Admin



Picture 3 Design Interface for Librarian

Chapter 4: System Development and Future Directions

4.1 Project Summary

The Library Management System project represents a comprehensive application of Java technology and software engineering principles to address the operational challenges faced by contemporary library environments. Through systematic analysis of library management requirements and strategic implementation of object-oriented programming concepts, the project has successfully delivered a functional desktop application that automates core library operations while maintaining data integrity and user experience standards appropriate for academic environments.

The development process encompassed the complete software engineering lifecycle, from initial requirements analysis through final system testing and documentation. The project team employed iterative development methodologies that facilitated continuous refinement of system components while ensuring alignment with established library management best practices. The resulting system demonstrates successful integration of database management capabilities, user interface design principles, and business logic implementation that collectively address the fundamental operational needs of library administrators and staff.

The system architecture implements a layered approach that separates presentation logic from business operations and data persistence mechanisms, creating a maintainable and extensible foundation for future enhancements. The application successfully demonstrates the practical application of theoretical concepts studied throughout the Java Technology course while providing tangible value to the library management domain. The project outcomes validate the effectiveness of object-oriented design patterns and demonstrate the capacity of Java technology to support complex business applications requiring robust data management and user interaction capabilities.

The successful completion of the Library Management System project has resulted in a robust, multi-faceted web application that addresses all core operational requirements of a modern library. The system is architected as a series of interconnected modules, each delivering specific, high-value functionality. The key deliverables and functional workflows are detailed below.

4.2 Achievements and Deliverables

A. User Authentication and Authorization Module

A secure and granular authentication system was implemented to manage access for all user types.

1. **Registration Workflow:** Prospective members can create an account by providing essential details (e.g., name, email, password). The system performs real-time input validation, and upon successful submission, a new user record is created with a default 'Member' role.
2. **Login Mechanism:** Registered users can access the system using their credentials. The backend verifies the password against a securely stored hash. Upon successful authentication, a session is initiated, and the user is granted access based on their assigned role.
3. **Role-Based Access Control (RBAC):** The system implements a strict RBAC model to differentiate user capabilities. Upon login, the application identifies the user's role (e.g., Member, Librarian, Administrator) and dynamically renders the appropriate interface and permissions, ensuring users can only access authorized functions.
4. **Secure Credential Storage:** To ensure user data integrity, all passwords are not stored in plaintext. Instead, they are processed through a one-way, salted hashing algorithm (e.g., bcrypt), mitigating the risk of unauthorized access from database breaches.
5. **Persistent Login:** A "Remember Me" feature has been incorporated, which generates a secure, long-lived token stored on the client-side, allowing for seamless re-authentication across sessions.
6. **Password Recovery:** A self-service password reset function is available. Users can request a password reset via their registered email address. The system then emails a unique, time-sensitive link that allows them to securely set a new password.

B. Core System Interface and Features

The primary user-facing components are designed for intuitive navigation and efficient information retrieval.

1. **Library Information Portal:** A public-facing page provides essential information, including the library's contact details, operating hours, address, and an embedded map for directions.
2. **Dynamic Book Catalog:** The system displays a comprehensive list of all available books in an interactive interface.
   - **Multi-Criteria Filtering:** Users can dynamically refine the book list using multiple filters simultaneously, such as genre, author, publication year, and availability status.

  - **Advanced Search Functionality:** A powerful search engine allows users to query the catalog based on various fields, including title, author, and ISBN, yielding precise and relevant results.

C. Administrative Management Console

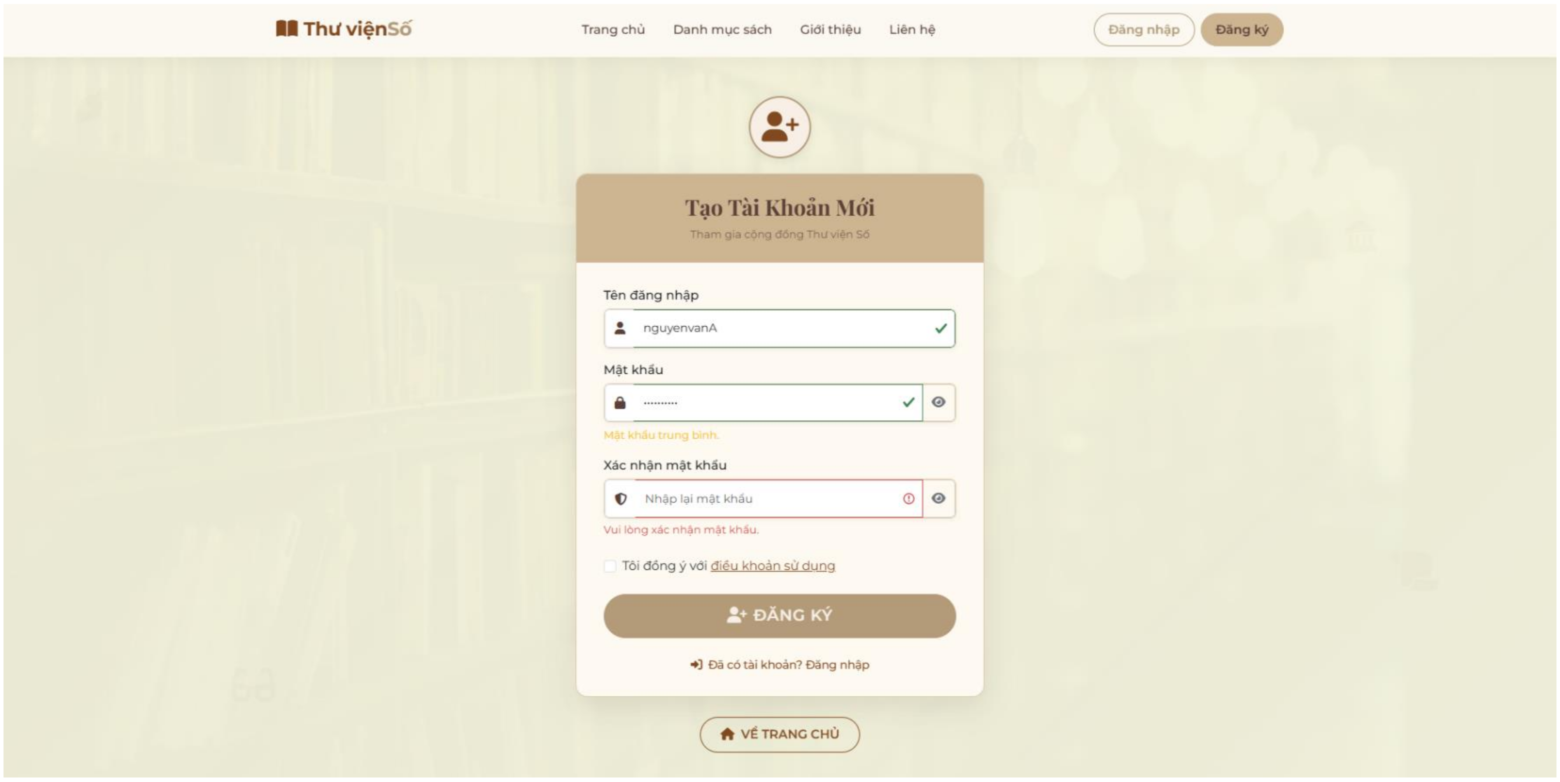
A dedicated backend provides administrators with complete control over the library's operations and data.

1. **Centralized Dashboard:** The admin dashboard offers a high-level overview of key metrics and provides quick access to all management functions.
2. **CRUD Functionality:** Full Create, Read, Update, and Delete (CRUD) capabilities have been implemented for all major data entities:
    - **User Management:** Administrators have full control over user accounts, including the ability to create, view, modify roles, and delete users.
3. **Announcement System:** A versatile communication tool has been developed for targeted messaging.
    - **General Announcements:** Administrators can publish system-wide notifications (e.g., holiday closures, events) that are visible to all users.
    - **Role-Specific Announcements:** The system allows for sending targeted messages visible only to users with a specific role (e.g., policy updates for 'Librarian' role).
4. **Statistical Reporting:** The system can generate and visualize key performance indicators, such as most borrowed books, user registration trends, and peak activity hours, providing valuable insights for strategic planning.

D. Support and Engagement

Integrated Chatbot: An automated chatbot is integrated into the user interface to provide 24/7 support. It is programmed to handle frequently asked questions (FAQs), assist users in locating books, and provide general information about library services, thereby enhancing user engagement and reducing the support load on staff.

The registration function integrates comprehensive security features for users while supporting the storage of registration information to enhance user convenience during the login process.



Tạo Tài Khoản Mới

Tham gia cộng đồng Thư viện Số

Tên đăng nhập: nguyenvanA

Mật khẩu: ··········

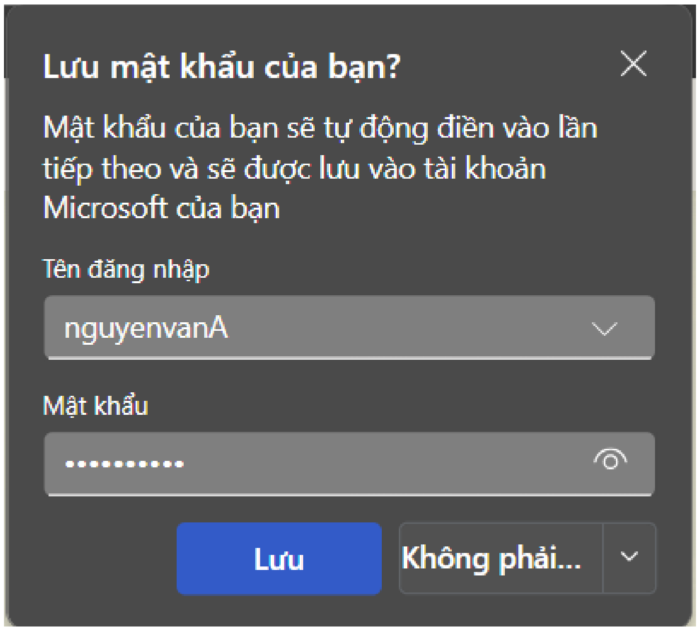
Mật khẩu trung bình.

Xác nhận mật khẩu: Nhập lại mật khẩu

Vui lòng xác nhận mật khẩu.

Tôi đồng ý với điều khoản sử dụng

Picture 4 Register



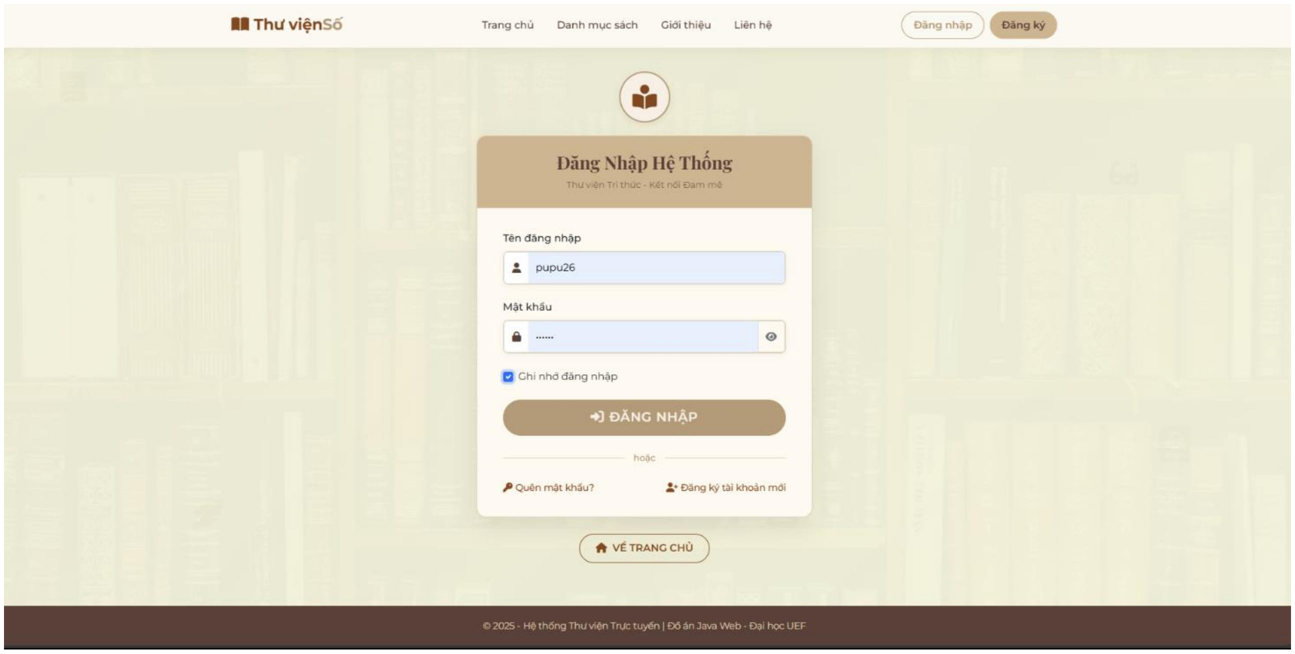
Lưu mật khẩu của bạn?

Mật khẩu của bạn sẽ tự động điền vào lần tiếp theo và sẽ được lưu vào tài khoản Microsoft của bạn

Tên đăng nhập: nguyenvanA

Mật khẩu: ••••••••••

Picture 5 Save Information for Login Function



Đăng Nhập Hệ Thống

Thư viện Tri thức - Kết nối Đam mê

Tên đăng nhập: pupu26

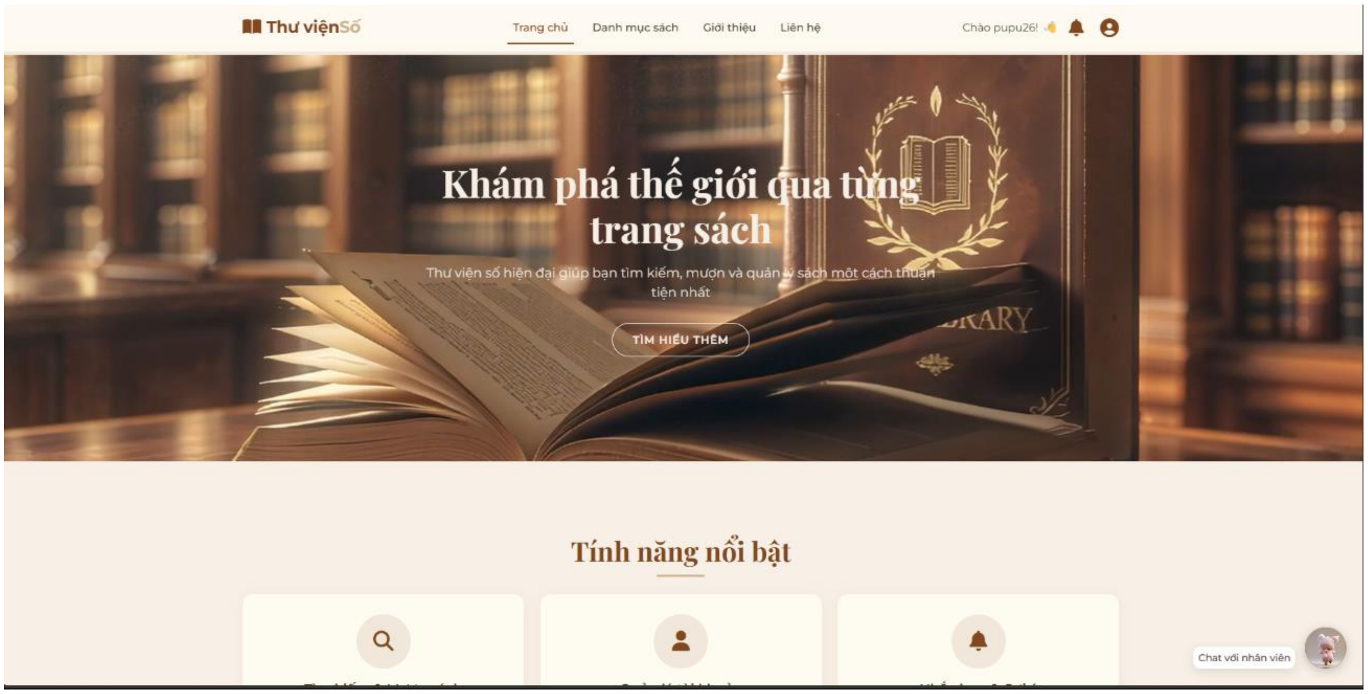
Mật khẩu: ••••••

[✓] Ghi nhớ đăng nhập

© 2025 - Hệ thống Thư viện Trực tuyến | Đồ án Java Web - Đại học UEF

Picture 6 Login

The system provides one-click login capability when passwords have been backed up, and includes password recovery support for users who have forgotten their credentials.



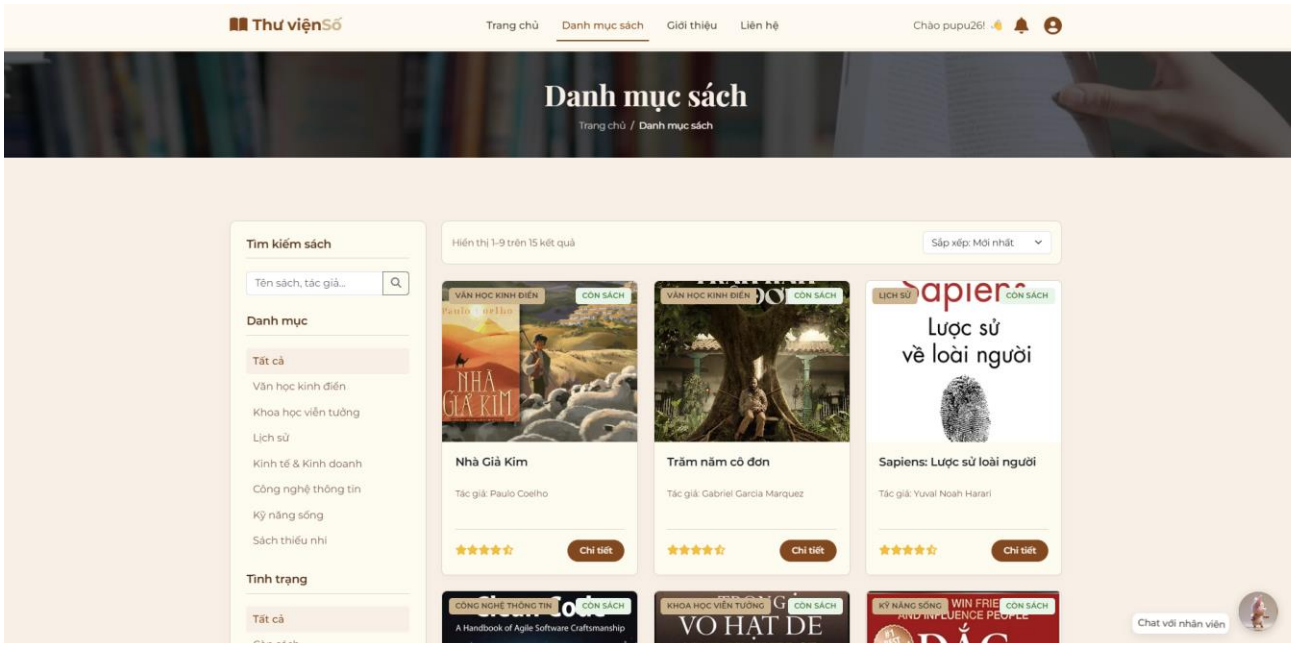
Khám phá thế giới qua từng trang sách

Thư viện số hiện đại giúp bạn tìm kiếm, mượn và quản lý sách một cách thuận tiện nhất

Tính năng nổi bật

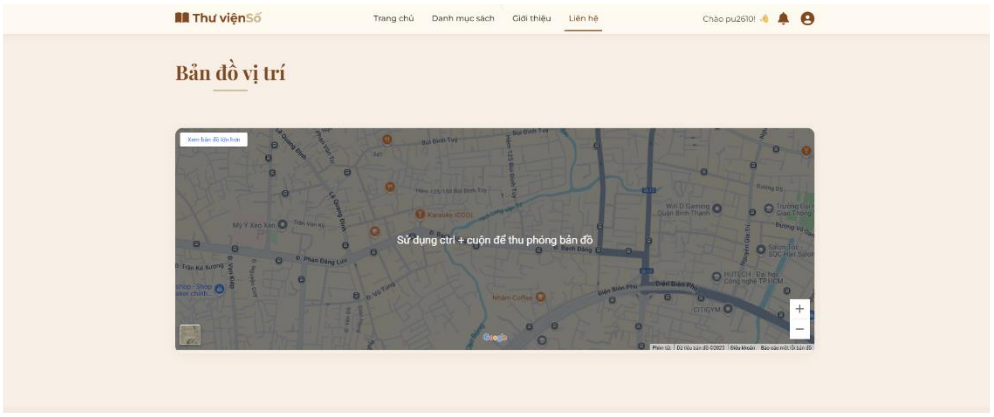
Picture 7 Home Dashboard

Upon successful login, the system displays the homepage interface, which contains essential information about the library that the web platform supports and manages.



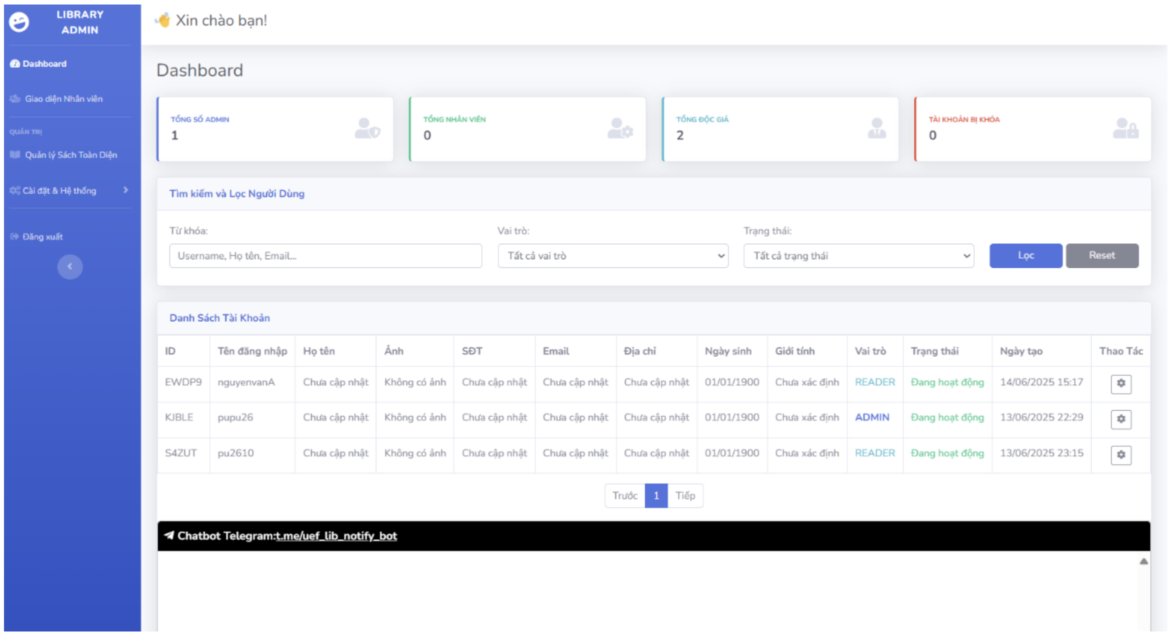
Picture 8 Book Catalog

The book catalog function presents a comprehensive list of available books, supplemented by intelligent search capabilities and advanced filtering options based on multiple criteria.



Picture 9 Map

The mapping function enables users to locate the precise geographical position of the library facility.



Xin chào bạn!

Dashboard

TỔNG SỐ ADMIN 1

TỔNG NHÂN VIÊN 0

TỔNG ĐỘC GIẢ 2

TÀI KHOẢN BỊ KHÓA 0

Tìm kiếm và Lọc Người Dùng

Từ khóa: Username, Họ tên, Email...

Vai trò: Tất cả vai trò

Trạng thái: Tất cả trạng thái

Lọc Reset

Danh Sách Tài Khoản

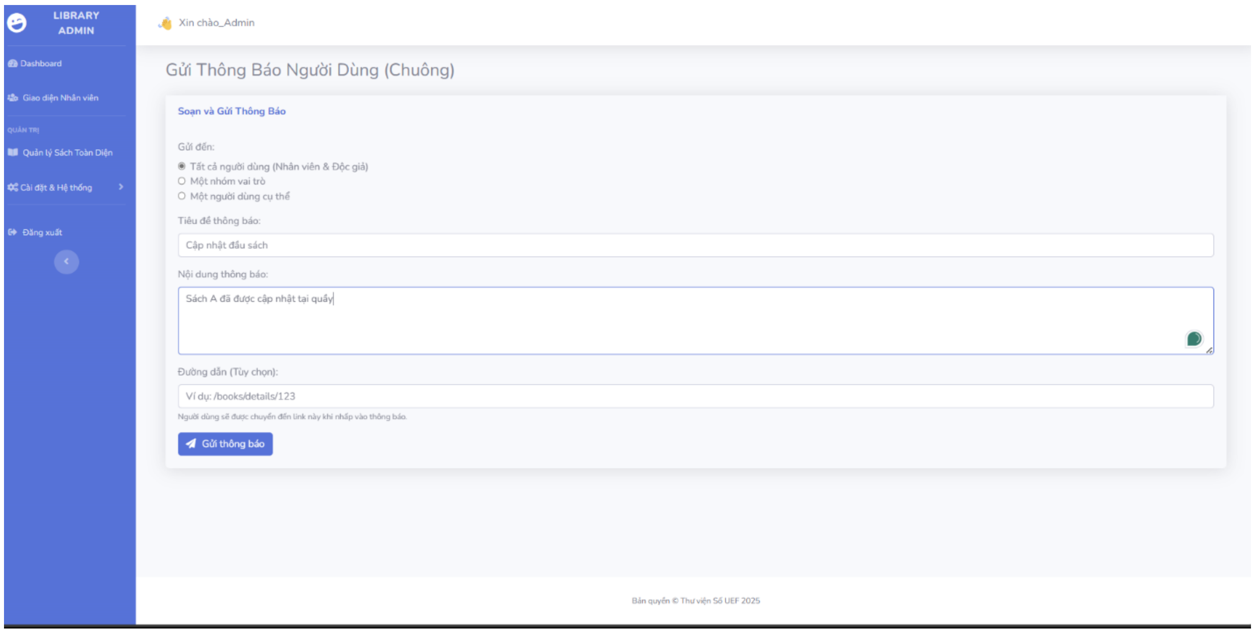
ID	Tên đăng nhập	Họ tên	Ảnh	SĐT	Email	Địa chỉ	Ngày sinh	Giới tính	Vai trò	Trạng thái	Ngày tạo	Thao Tác
EWDP9	nguyenvanA	Chưa cập nhật	Không có ảnh	Chưa cập nhật	Chưa cập nhật	Chưa cập nhật	01/01/1900	Chưa xác định	READER	Đang hoạt động	14/06/2025 15:17	⚙
KJBLE	pupu26	Chưa cập nhật	Không có ảnh	Chưa cập nhật	Chưa cập nhật	Chưa cập nhật	01/01/1900	Chưa xác định	ADMIN	Đang hoạt động	13/06/2025 22:29	⚙
S4ZUT	pu2610	Chưa cập nhật	Không có ảnh	Chưa cập nhật	Chưa cập nhật	Chưa cập nhật	01/01/1900	Chưa xác định	READER	Đang hoạt động	13/06/2025 23:15	⚙

Trước 1 Tiếp

Chatbot Telegram:t.me/uef_lib_notify_bot

Picture 10 Admin Dashboard

The administrative dashboard interface incorporates statistical data, filtering mechanisms, and comprehensive account listings with full CRUD (Create, Read, Update, Delete) functionality.



Xin chào_Admin

Gửi Thông Báo Người Dùng (Chuông)

Soạn và Gửi Thông Báo

Gửi đến:

- [✓ Tất cả người dùng (Nhân viên & Độc giả)]
- Một nhóm vai trò
- Một người dùng cụ thể

Tiêu đề thông báo: Cập nhật đầu sách

Nội dung thông báo: Sách A đã được cập nhật tại quầy

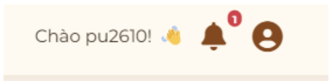
Đường dẫn (Tùy chọn): Ví dụ: /books/details/123

Người dùng sẽ được chuyển đến link này khi nhấp vào thông báo.

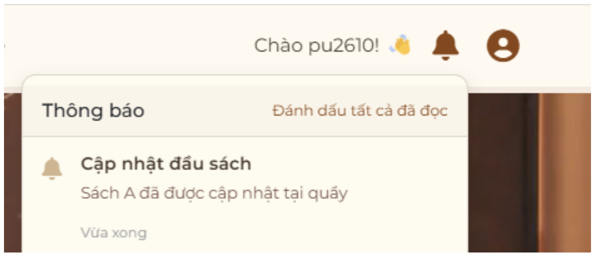
Gửi thông báo

Bản quyền © Thư viện Số UEF 2025

Picture 11 Notification



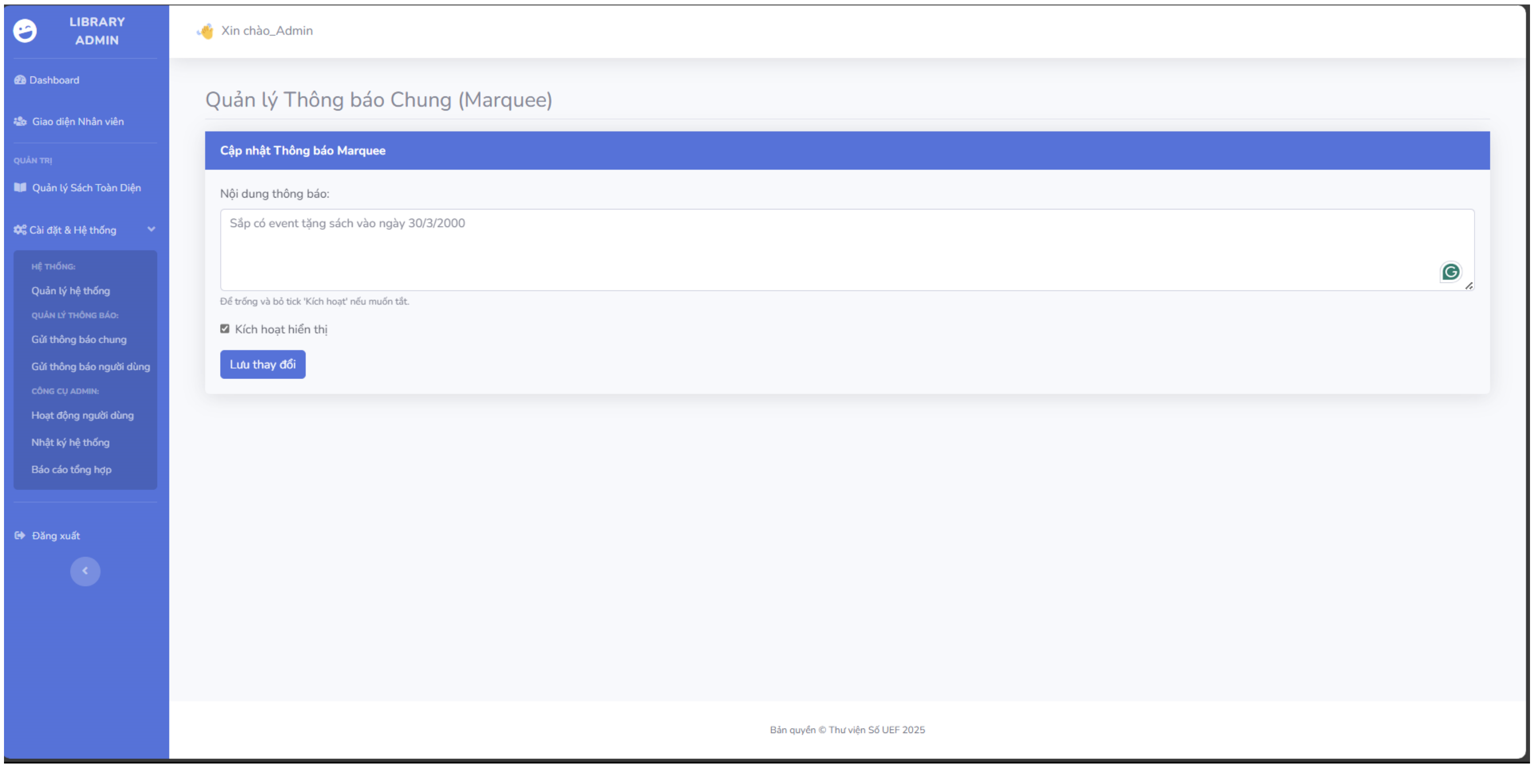
Picture 12 Notification Popup



Thông báo
Đánh dấu tất cả đã đọc
Cập nhật đầu sách
Sách A đã được cập nhật tại quầy
Vừa xong

Picture 13 Notification Details

The user notification system operates such that when administrators dispatch notifications, these alerts appear as popup notifications on the user's bell icon interface, targeting the intended recipients.



Quản lý Thông báo Chung (Marquee)

Cập nhật Thông báo Marquee

Nội dung thông báo:
Sắp có event tặng sách vào ngày 30/3/2000

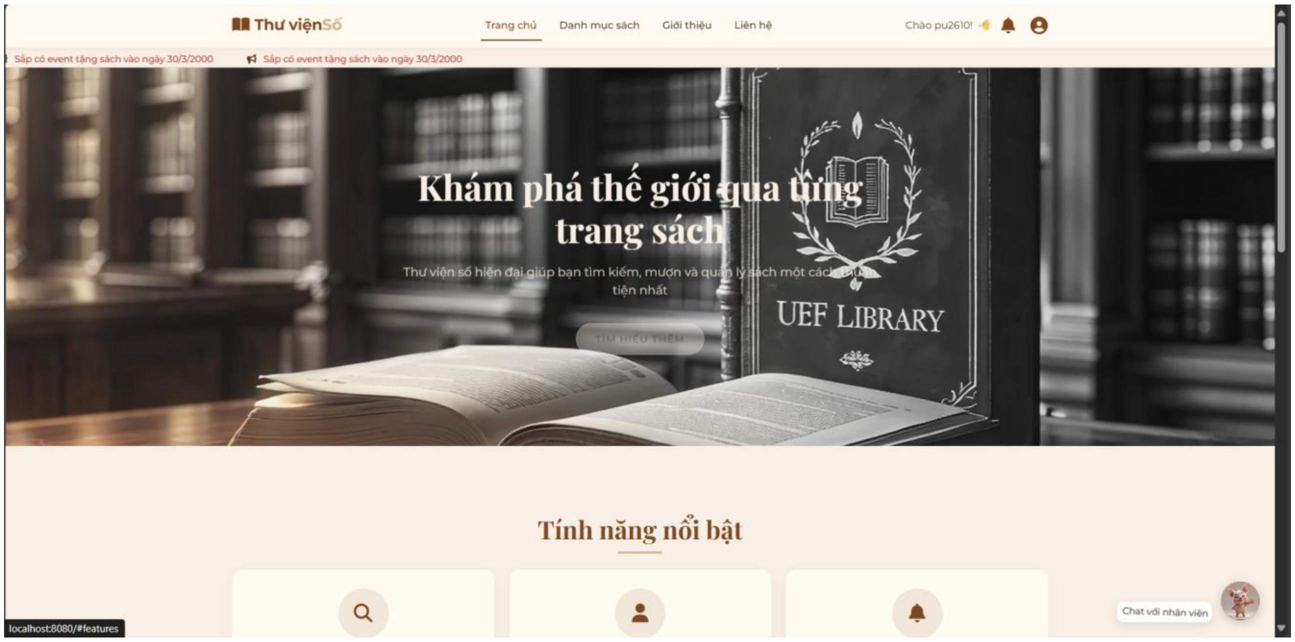
Để trống và bỏ tick 'Kích hoạt' nếu muốn tắt.

☑ Kích hoạt hiển thị

Lưu thay đổi

Bản quyền © Thư viện Số UEF 2025

Picture 14 General Notification


Thư việnSố
Trang chủ
Danh mục sách
Giới thiệu
Liên hệ
Chào pu2610!
Sắp có event tặng sách vào ngày 30/3/2000
Sắp có event tặng sách vào ngày 30/3/2000
Khám phá thế giới qua từng trang sách
Thư viện số hiện đại giúp bạn tìm kiếm, mượn và quản lý sách một cách thuận tiện nhất
TÌM HIỂU THÊM
UEF LIBRARY
Tính năng nổi bật
Chat với nhân viên
localhost:8080/#features

Picture 15 General Notification Result

The general announcement function ensures that when administrators publish announcements, these messages display as scrolling text notifications across the screen interface.

4.3 Challenges and Solutions

The development of the Library Management System presented numerous technical and methodological challenges that required innovative problem-solving approaches and collaborative team effort to resolve effectively. The complexity of modeling real-world library operations within software constraints necessitated careful analysis of business requirements and creative application of object-oriented design patterns to achieve optimal system architecture.

Database design challenges emerged from the need to accommodate complex relationships between library entities while maintaining query performance and data integrity standards. The implementation of appropriate foreign key relationships and constraint mechanisms required extensive testing to ensure that business rules were properly enforced at the database level. The team addressed these challenges through iterative schema refinement and comprehensive testing of edge cases that validated constraint effectiveness under various operational scenarios.

User interface design presented significant challenges in balancing functionality accessibility with screen space limitations inherent in desktop applications.

The need to present comprehensive information while maintaining intuitive navigation required careful consideration of information hierarchy and progressive disclosure techniques. The solution involved extensive user workflow analysis and iterative interface refinement based on usability testing feedback that identified optimal layout configurations for different operational tasks.

Concurrency and data consistency challenges arose from the need to ensure data integrity during simultaneous operations, particularly in lending and return processes where timing conflicts could result in inventory discrepancies. The implementation of appropriate transaction isolation levels and optimistic locking mechanisms provides robust protection against data corruption while maintaining system performance standards. The team addressed these challenges through comprehensive testing of concurrent access scenarios and implementation of appropriate error handling mechanisms.

Integration challenges between system components required careful attention to interface design and data flow management to ensure seamless operation across presentation, business logic, and data access layers. The solution involved implementation of standardized communication protocols between system components and comprehensive integration testing that validated proper data flow and error propagation throughout the system architecture.

4.4 Lessons Learned

The Library Management System development experience provided valuable insights into the practical application of software engineering principles and the complexities inherent in translating theoretical concepts into functional software solutions. The project demonstrated the critical importance of thorough requirements analysis in establishing realistic project scope and ensuring that system design decisions align with actual user needs and operational constraints.

The iterative development approach proved invaluable in managing project complexity while maintaining flexibility to accommodate design modifications discovered during implementation phases. The experience highlighted the benefits of early prototyping and user feedback integration in identifying usability issues and functional gaps that might not be apparent during initial design phases. The team learned that successful software development requires continuous balance between theoretical best practices and practical implementation constraints imposed by technology limitations and project timelines.

Database design and implementation experience reinforced the importance of careful schema planning and the challenges associated with balancing normalization principles with query performance requirements. The project demonstrated that effective database design requires understanding of both theoretical concepts and practical usage patterns to achieve optimal system performance. The team gained valuable experience in SQL query optimization and learned to anticipate performance bottlenecks through careful analysis of expected data access patterns. User interface development provided insights into the complexity of creating intuitive user experiences that accommodate diverse user capabilities and preferences. The experience highlighted the importance of user-centered design principles and the value of iterative refinement based on actual user feedback. The team learned that effective interface design requires careful consideration of cognitive load, workflow efficiency, and error prevention strategies that may not be immediately apparent during initial design phases. The testing and quality assurance process demonstrated the critical importance of comprehensive validation strategies that address both functional correctness and edge case scenarios. The experience highlighted the value of systematic testing approaches and the challenges associated with achieving adequate test coverage while maintaining development schedule constraints. The team learned that effective testing requires careful planning and integration throughout the development lifecycle rather than treatment as a separate post-development activity.

4.5 Future Enhancements

The Library Management System architecture provides a robust foundation for significant enhancements that would substantially expand operational capabilities and address evolving technological requirements in modern library environments. The proposed enhancement framework encompasses advanced technological integrations that leverage artificial intelligence, automation, and multilingual support to create a comprehensive library management ecosystem.

The implementation of **multilingual interface support** represents a critical enhancement for diverse academic environments, enabling seamless language switching capabilities that accommodate international users and multilingual institutional requirements. This enhancement would significantly expand the system's accessibility and usability across different cultural and linguistic contexts, supporting institutional diversity initiatives while maintaining operational consistency.

The integration of **automated email notification systems** would revolutionize patron communication through intelligent delivery of due date reminders and overdue notifications, reducing administrative overhead while improving patron compliance with borrowing policies.

The system would incorporate **automated reminder scheduling** that proactively manages overdue materials through systematic email campaigns, ensuring timely returns while minimizing manual intervention requirements.

Advanced **statistical reporting capabilities** would provide comprehensive analytics for borrowing trend analysis and collection status monitoring, enabling data-driven decision making for collection development and resource allocation strategies. These analytical tools would support strategic planning initiatives through detailed usage pattern recognition and predictive analytics that inform operational optimization efforts.

The incorporation of **barcode and QR code management systems** would modernize catalog processing through automated scanning and identification capabilities, significantly reducing manual data entry requirements while improving accuracy and efficiency in book processing workflows. This technological enhancement would align the system with contemporary library automation standards and support rapid catalog expansion capabilities.

Image management functionality for book covers would enhance user experience through visual catalog browsing capabilities while supporting modern patron expectations for multimedia catalog interfaces. The system would accommodate high-resolution image storage and display optimization that maintains performance standards while providing engaging visual navigation options.

The implementation of **comprehensive activity monitoring systems** would provide detailed user behavior tracking and system audit capabilities, supporting security requirements while enabling operational analysis through detailed logging and reporting mechanisms. These monitoring capabilities would ensure compliance with institutional governance standards while providing valuable operational insights.

The integration of **intelligent virtual search assistance** represents a transformative enhancement that would incorporate natural language processing capabilities for contextual search understanding and sophisticated query interpretation. This artificial intelligence integration would significantly improve resource discovery efficiency while accommodating varied user search methodologies and preferences.

Intelligent question-answering capabilities would provide automated responses to inquiries regarding book content, author information, and collection details, reducing staff workload while improving patron service accessibility. The system would leverage machine learning algorithms to continuously improve response accuracy and expand knowledge base capabilities through usage pattern analysis.

Automatic translation functionality would enable real-time translation of book information and interface elements across multiple languages, supporting international collections and diverse user populations while maintaining consistent system functionality regardless of language preferences. This enhancement would significantly expand the system's global applicability and institutional value.

The incorporation of **intelligent content quality assessment** would analyze patron reviews and feedback to determine content quality ratings, supporting collection development decisions through automated evaluation of resource value and patron satisfaction metrics. This analytical capability would enable systematic quality management while providing valuable insights for acquisition strategies and collection optimization initiatives.

4.6 Recommendations

The successful deployment and maintenance of the Library Management System requires careful consideration of implementation strategies, staff training requirements, and ongoing support mechanisms that ensure optimal system utilization and user satisfaction. The recommendation framework addresses both immediate deployment considerations and long-term system evolution strategies that maximize return on development investment while maintaining operational effectiveness.

Deployment planning should prioritize comprehensive staff training that addresses both technical system operation and workflow integration strategies. The training program should accommodate varying levels of technological proficiency among library personnel while ensuring that all users can effectively utilize system capabilities relevant to their operational responsibilities. The development of comprehensive user documentation and quick reference guides would support ongoing system utilization and reduce dependency on technical support resources.

Data migration strategies require careful planning to ensure accurate transfer of existing library records into the new system while maintaining historical transaction data essential for operational continuity. The implementation of parallel operation periods would enable gradual transition from existing systems while providing fallback capabilities during initial deployment phases. Comprehensive backup and recovery procedures should be established before system deployment to protect against data loss and ensure business continuity.

System maintenance and support strategies should establish clear procedures for bug reporting, feature requests, and system updates that maintain system reliability while accommodating evolving operational requirements. The implementation of regular system monitoring and performance evaluation procedures would identify potential issues before they impact operational effectiveness.

The establishment of user feedback mechanisms would provide ongoing input for system refinement and enhancement priorities.

Future development planning should consider the potential for system expansion and integration with emerging library technologies while maintaining compatibility with existing operational procedures. The evaluation of web-based migration strategies should begin early in the system deployment lifecycle to ensure that future enhancements align with institutional technology strategies and user expectations. The development of formal system documentation and code review procedures would facilitate future enhancement projects while maintaining system stability and reliability standards.

CONCLUSION

This Library Management System project successfully demonstrates the practical application of Java technology and object-oriented programming to solve real-world institutional challenges. The developed desktop application effectively automates library operations through integrated book cataloging, member management, and lending processes, replacing manual systems with reliable digital solutions.

The project achieves its core objectives by implementing a robust three-tier architecture that ensures clear separation between presentation, business logic, and data layers. The user interface balances comprehensive functionality with intuitive design, accommodating users with varying technical skills. Advanced Java features including JDBC connectivity, error handling, and input validation ensure data integrity and system reliability.

Comprehensive testing validated system functionality across multiple scenarios, while development challenges in database optimization and UI refinement provided valuable learning experiences. The system establishes a solid foundation for future enhancements including AI integration and advanced analytics.

This project validates systematic software engineering approaches in addressing institutional needs, contributing meaningfully to educational digitization and modern library management practices.