LIBRARY MANAGEMENT SYSTEM

Abdullah Sajid
Faizan Ali
M. Saim

AF-IAST
Skilling Pakistan

CERTIFICATION

This is to certify that the project entitled LibraSys (Library Management System) is completed and submitted by Abdullah Sajid, Faizan Ali and M. Saim Ahmed from BS(CS) 5th semester at Pak-Austria Fachhochschule - Institute of Applied Sciences and Technology (PAF-IAST), as part of the Software Engineering course end semester project taught by Sir Mr. Abdul Basit Shah Kazmi.

S.No	Name	Reg No
1.	Abdullah Sajid	B21F0060CS003
2.	Faizan Ali	B21F0243CS016
3.	M. Saim Ahmed	B21F0248CS017

CLASS OF 2021
THE
AFS
GROUP
PROJECT GROUP STAMP

Dated: 17th Jan, 2024

AF-IAST
Skilling Pakistan

Pak-Austria Fachhochschule
Institute of Applied Sciences and Technology

Submitted To	:	**Mr. Abdul Basit Shah Kazmi**
		Abdullah Sajid
Submitted By	:	**Faizan Ali**
		M. Saim
Course Title	:	**Software Engineering**
Department	:	**IT & CS**
Submission Date	:	**17th Jan, 2024**

Marks	
Remarks	

TABLE OF CONTENT:

Contents

PROBLEM SCENERIO:

Imagine a bustling day in the University Library Management System where students, faculty, staff, and librarians engage with the system to facilitate various library-related tasks. New students eager to explore the vast collection of resources register with the library, providing necessary information to create their accounts. Throughout the day, users actively search for books, leveraging the system's functionalities to locate materials based on titles, authors, or genres.

Upon finding desired books, students and faculty seamlessly borrow them by scanning their library cards and the book's barcode. To accommodate their needs, users also have the option to renew borrowed books online, ensuring continued access to valuable resources. Returning borrowed books is a straightforward process facilitated by the system, updating the book's status to "available" upon check-in.

User profiles remain dynamic as individuals manage their information through profile updates, including changes to contact details and password modifications. The system provides an avenue for users to voice their thoughts on library services by submitting feedback through an online form. Users authenticate themselves to access their accounts, ensuring a secure and personalized experience within the library system.

The library administrators, represented by librarians, play a crucial role in managing the system. They handle fines associated with overdue books, reviewing and updating fine records as needed. Librarians actively engage with users by sending notifications, reminding them to return overdue books and notifying them when reserved books become available.

In the background, the system generates reports to assist librarians in understanding book circulation patterns, popular genres, and outstanding fines. Librarians also review and respond to user feedback received through the online form, contributing to a continuous improvement process. Maintenance of the library catalog is an ongoing task for librarians, involving activities such as adding new books, removing outdated ones, and updating existing book details.

As part of the system's robust operations, the librarians have the capability to update fine details for specific users based on valid reasons for adjustments. Additionally, the system automates the process of sending notifications to users with overdue books, ensuring timely communication. Users receive notifications not only for overdue books but also when reserved books become available or when a reservation is canceled.

This vibrant scenario encompasses a myriad of interactions, showcasing the diverse functionalities of the University Library Management System. Students and faculty seamlessly engage with the system to access and manage resources, while librarians efficiently administer the system's operations, ensuring a smooth and user-friendly experience for all.

CHOSSING THE SDLC MODEL:

SDLC Model:

The Software Development Life Cycle (SDLC) model is a structured approach to software development that provides a framework for planning, creating, testing, deploying, and maintaining an information system. It encompasses various phases, including planning, design, development, testing, deployment, and maintenance.

Applying Scrum Model to Library Management System:

In the context of developing a Library Management System, the Scrum model is adopted due to its iterative and incremental nature. This model allows the development team to adapt to changing requirements and incorporate continuous feedback from stakeholders. The Scrum model's emphasis on collaboration, transparency, and delivering incremental value aligns well with the dynamic nature of library management systems.

Scrum Roles:

1. **Product Owner:** Represents stakeholders, prioritizes and manages the product backlog, and is responsible for defining and communicating product requirements.
2. **Scrum Master:** Facilitates the Scrum process, ensures the team follows Scrum practices, and helps the team overcome challenges and impediments.
3. **Development Team:** A cross-functional team responsible for delivering increments of potentially releasable functionality. Collaborates on user stories during sprint planning and commits to delivering them.

Scrum Artifacts:

1. **Product Backlog:** A list of all functional and non-functional requirements prioritized by the Product Owner based on user value and project goals.
2. **Sprint Backlog:** A subset of the product backlog selected for a specific sprint, containing user stories and tasks committed to by the development team.
3. **Increment:** The sum of all completed user stories and features at the end of a sprint, forming a potentially releasable product.

Scrum Ceremonies:

The Scrum ceremonies play a crucial role in the iterative and adaptive nature of the Scrum model.

Sprint Planning Meeting (2 weeks): During this phase, the Product Owner takes the lead by presenting the highest-priority items from the product backlog. The Development Team actively engages by selecting user stories and collectively defining the sprint goal, establishing a clear direction for the upcoming two weeks.

Daily Stand-up Meetings (15 minutes each): A daily ritual within the sprint, these brief stand-up meetings serve as a platform for team members to provide updates on their progress, discuss plans, and highlight any challenges encountered. Facilitated by the Scrum Master, these meetings ensure constant communication and alignment within the development team.

Sprint Review (1 week): Conducted at the conclusion of each sprint, the Sprint Review is a pivotal ceremony where the Development Team showcases the completed increment to stakeholders. This interactive session allows stakeholders to provide valuable feedback, fostering a collaborative environment. The Product Owner, in turn, updates the product backlog based on this input, ensuring alignment with evolving requirements.

Sprint Retrospective (1 week): At the end of each sprint, the team engages in a reflective Sprint Retrospective. This session involves an analysis of the sprint, identifying successes and areas for improvement. The team collectively explores what went well, what could be enhanced, and formulates actionable strategies for continuous improvement in subsequent sprints.

TIMELINE:

Sprint 1-2 (8 weeks):

1. Focus on high-priority features like User Registration, Book Search, Retrieval, and Book Borrowing.
2. Incremental development and frequent feedback.

Sprint 3-4 (7 weeks):

1. Emphasis on Renewal, Return functionalities, and Book History View.
2. Continuous collaboration and adjustments based on user feedback.

Sprint 5-6 (6 weeks):

1. Prioritize Profile Management, Feedback Submission, User Authentication, and Fine View.
2. Regular refinement of the product backlog.

Sprint 7-8 (4 weeks):

1. Concentrate on functionalities like Cancel Membership, Fine Management, Notification Sending, and Report Generation.
2. Iterative development and adjustments based on stakeholder feedback.

Sprint 9 (2 weeks):

1. Focus on Review Feedback, Maintain Book Records, and Address Update Fine Details.
2. Continuous refinement of backlog items.

Sprint 10 (3 weeks):

1. Implement automated notifications, address remaining non-functional requirements, and finalize system integration and testing.

Sprint 11 (2 weeks):

1. Prepare for User Acceptance Testing (UAT) and deployment.
2. Final adjustments based on UAT feedback.

Sprint 12 (1 week):

1. Release for live use.
2. Post-deployment support and maintenance.

Explanation of SDLC Choice:

The Scrum model is chosen for its iterative and incremental approach, allowing the development team to adapt to changing requirements and incorporate continuous feedback from stakeholders. Scrum's emphasis on collaboration, transparency, and delivering incremental value aligns well with the dynamic nature of library management systems. This approach fosters a responsive and adaptive development process, ensuring that the final product meets user expectations and addresses evolving needs effectively.

REQUIREMENT ELICITATION:

1. FUNCTIONAL REQUIREMENTS:

User Registration:

The system must allow users (students and staff) to register for a library account, providing necessary information.

Forget Password:

- Users should have the option to recover or reset forgotten passwords.
- The system must facilitate a secure process for identity verification and password reset.

Logout:

- Users should be able to log out of the library system.
- The system must terminate the user's session upon logout.

Book Search and Retrieval:

The system should enable users to search for books based on various criteria (title, author, genre) and display relevant results.

Book Borrowing:

Users should be able to borrow books, and the system must update the book's status to "checked out" for the respective user.

Book Renewal:

The system should allow users to renew borrowed books, updating the due date accordingly.

Book Return:

Users should be able to return borrowed books, and the system must update the book's status to "available."

Profile Management:

Users should have the ability to manage their profiles, including updating contact information and changing passwords.

Feedback Submission:

The system should allow users to submit feedback on library services.

User Authentication:

The system must authenticate users during login using unique usernames and passwords.

Book History View:

Users should be able to view their borrowing history, showing previously borrowed books.

Fine View:

Users should be able to view information about fines related to overdue books.

Cancel Membership:

Librarians should be able to cancel a user's library membership.

Fine Management:

Librarians should have the ability to manage fines for overdue books, including viewing fine history.

Notification Sending:

Librarians should be able to send notifications to users (e.g., overdue book notifications, reservation available notifications).

Report Generation:

Librarians should have the ability to generate reports on library statistics (e.g., circulation, popular genres).

Review Feedback:

Librarians should be able to review and respond to user feedback.

Maintain Book Records:

Librarians should have the ability to add, remove, and update books in the library catalog.

Update Fine Details:

Librarians should be able to update fine details for specific users.

Overdue Notification:

The database system should automatically send notifications for overdue books.

Reservation Available Notification:

The database system should automatically send notifications when reserved books become available.

Reservation Cancellation Notification:

The database system should automatically send notifications when a reservation is canceled.

2. NON - FUNCTIONAL REQUIREMENTS:

Performance:

- The system should be capable of handling concurrent users, ensuring smooth performance even during peak times.
- Book search and retrieval operations should take no more than 3 seconds.

Scalability:

The system must be scalable to accommodate a growing library collection and user base.

Reliability:

- The system must have a 99.9% uptime.
- Data backup should occur nightly, with quick restoration in case of system failures.

Security:

- User data, especially passwords, must be securely stored using encryption.
- Access to sensitive functions (e.g., book removal) is restricted to authorized roles.

Usability:

- The user interface must be intuitive and accessible for users of all technical levels.
- Navigation should be clear, and common tasks should be easily executable.

Compatibility:

The system must be compatible with common web browsers (Chrome, Firefox, Safari).

Data Integrity:

Ensure data consistency and integrity, especially during simultaneous check-in/check-out operations.

Compliance:

The system must comply with data protection regulations (e.g., GDPR, HIPAA).

Audit Trail:

Maintain an audit trail for critical actions, such as book removal or user role changes.

Accessibility:

The system must be accessible to users with disabilities, following WCAG guidelines.

These requirements cover a range of functionalities and characteristics that are crucial for the successful development and operation of the Library Management System. Adjustments can be made based on specific project requirements and system design considerations.

SOFTWARE REQUIREMENT SPECIFICATION:

1. Introduction

1.1 Purpose

The purpose of the Library Management System (LibraSys) is to streamline and automate library operations, providing an efficient and user-friendly platform for both librarians and members. LibraSys aims to enhance the overall library experience by automating tasks such as book search, borrowing, returning, and fine management.

1.2 Scope

LibraSys encompasses user registration, book management, and interactions between members and librarians. It aims to provide a centralized system for maintaining accurate and up-to-date information on available books, member accounts, and related transactions.

1.3 Definitions, Acronyms, and Abbreviations

LibraSys: Integrated Library Management System.

UC: Use Case.

SRS: Software Requirements Specification.

API: Application Programming Interface.

1.4 References

ISBN Database: For retrieving detailed book information.

User Authentication Standards: For ensuring secure member logins.

Data Protection Laws: To adhere to legal requirements in handling user data.

1.5 Overview of the Document

This document serves as a comprehensive guide for stakeholders involved in the development and use of LibraSys. It outlines the system's purpose, scope, definitions, and references, providing a foundational understanding for successful system development and operation.

2. Overall Description

2.1 Product Perspective

LibraSys is a standalone system that interacts with external databases, such as the ISBN Database for book details. It operates in a networked environment, allowing members and librarians to access the system through web-based interfaces.

2.2 Product Features

Key features of LibraSys include user registration, book search, borrowing and returning books, fine management, notifications, reports, and feedback handling. The system is designed to support the needs of both members and librarians efficiently.

2.3 User Classes and Characteristics

LibraSys caters to two primary user classes: members and librarians. Members are individuals who use library services, while librarians are responsible for managing the system, performing administrative tasks, and maintaining book records.

2.4 Operating Environment

LibraSys operates in a web-based environment, supporting common browsers such as Chrome, Firefox, and Safari. It requires a stable internet connection for optimal performance.

2.5 Design and Implementation Constraints

The system must adhere to industry-standard security protocols and comply with data protection regulations. Additionally, it should support scalability to accommodate a growing number of users and books.

2.6 User Documentation

Comprehensive user documentation, including manuals and guides, will be provided to assist members and librarians in understanding and using LibraSys effectively.

2.7 Assumptions and Dependencies

Assumptions:

Users possess basic internet literacy.

Members have valid email addresses for registration.

Dependencies:

Availability of a reliable internet connection for system access.

Compliance with data protection laws and regulations.

3.1 User Registration

3.1.1 Description

User registration involves capturing and storing member information in the system. This includes personal details, contact information, and login credentials.

3.1.2 Inputs

- Member details (name, address, contact information).
- Email address for communication.
- Desired login credentials (username and password).

3.1.3 Processing

- Validate input data for completeness and correctness.
- Generate a unique member ID.
- Store member information in the database.
- Send a confirmation email to the registered member.

3.1.4 Outputs

- Unique member ID.
- Confirmation email.

3.1.5 User Interface

- A user-friendly registration form with fields for personal details, contact information, and login credentials.

3.1.6 Performance

- Response time for registration: < 5 seconds.
- Concurrent user registrations: Up to 100 per minute.

3.1.7 Security

- Encrypt sensitive information during transmission.
- Store passwords securely using encryption.
- Implement account lockout mechanisms after multiple failed login attempts.

3.2 Book Search

3.2.1 Description

Book search allows members to find and access information about available books in the library.

3.2.2 Inputs

- Book title, author, or keyword.

3.2.3 Processing

- Query the database based on user input.
- Retrieve matching book records.
- Display results to the user.

3.2.4 Outputs

- List of books matching the search criteria.

3.2.5 User Interface

- A search bar on the member interface with options for filtering by title, author, or keyword.

3.2.6 Performance

- Response time for search: < 3 seconds.
- Concurrent searches: Up to 200 per minute.

3.2.7 Security

- Ensure user authentication before allowing access to search features.
- Implement secure communication protocols for transmitting search queries.

3.3 Borrow a Book

3.3.1 Description

Members can borrow a book after locating it through the search feature.

3.3.2 Inputs

- Member ID.
- Book ID.

3.3.3 Processing

- Verify member's eligibility to borrow.

- Update book status to "borrowed."
- Record transaction details.

3.3.4 Outputs

- Transaction receipt.
- Updated book availability status.

3.3.5 User Interface

- A "Borrow" button on the book details page.

3.3.6 Performance

- Response time for borrowing: < 5 seconds.
- Concurrent borrow transactions: Up to 50 per minute.

3.3.7 Security

- Verify member identity before processing the transaction.
- Log borrowing activities for audit trail purposes.

3.4 Renew a Book

3.4.1 Description

- Members can renew a borrowed book to extend the loan period.

3.4.2 Inputs

- Member ID.
- Book ID.

3.4.3 Processing

- Verify member's eligibility to renew.
- Update book's due date.
- Record renewal transaction details.

3.4.4 Outputs

- Transaction receipt.
- Updated due date for the book.

3.4.5 User Interface

- A "Renew" button on the member's borrowed books list.

3.4.6 Performance

- Response time for renewal: < 5 seconds.
- Concurrent renewal transactions: Up to 30 per minute.

3.4.7 Security

- Authenticate member before allowing renewal.
- Log renewal activities for audit trail purposes.

3.5 Return a Book

3.5.1 Description

- Members can return a borrowed book, updating the library's records.

3.5.2 Inputs

- Member ID.
- Book ID.

3.5.3 Processing

- Verify member's eligibility to return.
- Update book status to "available."
- Record return transaction details.

3.5.4 Outputs

- Transaction receipt.
- Updated book availability status.

3.5.5 User Interface

- A "Return" button on the member's borrowed books list.

3.5.6 Performance

- Response time for return: < 5 seconds.
- Concurrent return transactions: Up to 40 per minute.

3.5.7 Security

- Authenticate member before allowing the return.
- Log return activities for audit trail purposes.

3.6 Extend Pay Fine

3.6.1 Description

- Members can extend the deadline for paying fines accrued during book returns.

3.6.2 Inputs

- Member ID.
- Fine details.

3.6.3 Processing

- Verify member's eligibility to extend fine payment.
- Update fine payment deadline.
- Record fine extension transaction details.

3.6.4 Outputs

- Transaction receipt.
- Updated fine payment deadline.

3.6.5 User Interface

- An "Extend Fine" option on the member's fine details page.

3.6.6 Performance

- Response time for extending fine: < 5 seconds.
- Concurrent fine extension transactions: Up to 20 per minute.

3.6.7 Security

- Authenticate member before allowing fine extension.
- Log fine extension activities for audit trail purposes.

3.7 Profile Management

3.7.1 Description

- Members can manage their profiles, updating personal information and preferences.

3.7.2 Inputs

- Member ID.
- Updated personal information (name, address, contact details).

3.7.3 Processing

- Validate and update member information.
- Record profile update transaction details.

3.7.4 Outputs

- Transaction receipt.
- Updated member profile.

3.7.5 User Interface

- An "Edit Profile" option on the member's dashboard.

3.7.6 Performance

- Response time for profile update: < 3 seconds.

- Concurrent profile update transactions: Up to 30 per minute.

3.7.7 Security

- Authenticate member before allowing profile updates.
- Log profile update activities for audit trail purposes.

3.8 Book History

3.8.1 Description

- Members can view their borrowing history, providing a record of past transactions.

3.8.2 Inputs

- Member ID.

3.8.3 Processing

- Retrieve member's borrowing history from the database.
- Display the history to the member.

3.8.4 Outputs

- List of borrowed books with transaction details.

3.8.5 User Interface

- A "View History" option on the member's dashboard.

3.8.6 Performance

- Response time for history retrieval: < 5 seconds.
- Concurrent history retrieval transactions: Up to 25 per minute.

3.8.7 Security

- Authenticate member before allowing access to the borrowing history.
- Log history access for audit trail purposes.

3.9 Feedback

3.9.1 Description

- Members can provide feedback on their library experience, helping improve services.

3.9.2 Inputs

- Member ID.
- Feedback content and rating.

3.9.3 Processing

- Validate and store feedback.
- Analyze feedback for improvement insights.

3.9.4 Outputs

- Confirmation message.

3.9.5 User Interface

- A "Provide Feedback" option on the member's dashboard.

3.9.6 Performance

- Response time for feedback submission: < 3 seconds.
- Concurrent feedback submissions: Up to 15 per minute.

3.9.7 Security

- Authenticate member before allowing feedback submission.
- Anonymous feedback option available for privacy.

3.10 Cancel Membership

3.10.1 Description

- Members can request to cancel their library membership.

3.10.2 Inputs

- Member ID.
- Reason for cancellation.

3.10.3 Processing

- Verify member's request and reason.
- Cancel the membership.
- Record cancellation transaction details.

3.10.4 Outputs

- Confirmation message.
- Cancellation transaction details.

3.10.5 User Interface

- A "Cancel Membership" option on the member's dashboard.

3.10.6 Performance

- Response time for membership cancellation: < 5 seconds.
- Concurrent cancellation transactions: Up to 10 per minute.

3.10.7 Security

- Authenticate member before processing the cancellation request.
- Log membership cancellation activities for audit trail purposes

3.11 Maintain Book Records

3.11.1 Description

- Librarians can manage book records, including adding new books, updating information, and removing obsolete entries.

3.11.2 Inputs

- Librarian ID.
- Book details (title, author, ISBN, quantity, etc.).

3.11.3 Processing

- Validate and add/update book information in the database.
- Adjust available quantities accordingly.
- Remove books if necessary.
- Record book management transaction details.

3.11.4 Outputs

- Confirmation message.
- Updated book records.

3.11.5 User Interface

- A "Manage Books" option on the librarian's dashboard.

3.11.6 Performance

- Response time for book management: < 5 seconds.
- Concurrent book management transactions: Up to 20 per minute.

3.11.7 Security

- Authenticate librarian before allowing book management.
- Logbook management activities for audit trail purposes.

3.12 Fine

3.12.1 Description

- Librarians can manage fines, including updating fine amounts, marking fines as paid, and maintaining fine records.

3.12.2 Inputs

- Librarian ID.
- Member ID.
- Fine details.

3.12.3 Processing

- Verify librarian's eligibility to manage fines.
- Update fine details based on member payments.
- Record fine management transaction details.

3.12.4 Outputs

- Confirmation message.
- Updated fine records.

3.12.5 User Interface

- A "Manage Fines" option on the librarian's dashboard.

3.12.6 Performance

- Response time for fine management: < 5 seconds.
- Concurrent fine management transactions: Up to 15 per minute.

3.12.7 Security

- Authenticate librarian before allowing fine management.
- Log fine management activities for audit trail purposes.

3.13 Notifications

3.13.1 Description

- The system sends notifications to members for events such as due dates, reservations, and overdue books.

3.13.2 Inputs

- System-generated events.
- Member ID.
- Notification content.

3.13.3 Processing

- Trigger notifications based on predefined events.
- Send notifications to members via email or in-app messages.

3.13.4 Outputs

- Notification delivery reports.

3.13.5 User Interface

- A notification center in the member's dashboard.

3.13.6 Performance

- Response time for notification delivery: < 2 seconds.
- Concurrent notification deliveries: Up to 50 per minute.

3.13.7 Security

- Secure communication channels for delivering notifications.
- Log notification activities for audit trail purposes.

3.14 Reports

3.14.1 Description

- Generate reports for librarians to analyze trends, book popularity, and member activities.

3.14.2 Inputs

- Librarian ID.
- Report parameters (timeframe, category, etc.).

3.14.3 Processing

- Retrieve relevant data from the database.
- Generate reports based on specified parameters.
- Display reports to librarians.

3.14.4 Outputs

- Visual reports and data exports.

3.14.5 User Interface

- A "Generate Reports" option on the librarian's dashboard.

3.14.6 Performance

- Response time for report generation: < 8 seconds.
- Concurrent report generations: Up to 10 per minute.

3.14.7 Security

- Authenticate librarian before allowing report generation.
- Restrict access to sensitive data in reports.

3.15 Review Feedback

3.15.1 Description

- Librarians can review and respond to member feedback, addressing concerns and implementing improvements.

3.15.2 Inputs

- Librarian ID.
- Member feedback details.

3.15.3 Processing

- Retrieve and display member feedback.
- Allow librarians to respond to feedback.

3.15.4 Outputs

- Response messages to members.

3.15.5 User Interface

- A "Review Feedback" option on the librarian's dashboard.

3.15.6 Performance

- Response time for feedback review: < 5 seconds.
- Concurrent feedback reviews: Up to 15 per minute.

3.15.7 Security

- Authenticate librarian before allowing feedback review.
- Log feedback review activities for audit trail purposes.

3.16 Send Overdue Notification

3.16.1 Description

- Automatically send notifications to members with overdue books, reminding them to return or renew.

3.16.2 Inputs

- System-generated overdue events.
- Member ID.
- Notification content.

3.16.3 Processing

- Identify members with overdue books.
- Trigger and send overdue notifications.

3.16.4 Outputs

- Notification delivery reports.

3.16.5 User Interface

- A notification center in the member's dashboard.

3.16.6 Performance

- Response time for overdue notification delivery: < 2 seconds.
- Concurrent overdue notification deliveries: Up to 30 per minute.

3.16.7 Security

- Secure communication channels for delivering notifications.
- Log overdue notification activities for audit trail purposes.

3.17 Send Reservation Available Notification

3.17.1 Description

- Notify members when a reserved book becomes available for borrowing.

3.17.2 Inputs

- System-generated reservation available events.
- Member ID.
- Notification content.

3.17.3 Processing

- Identify members with available reserved books.
- Trigger and send reservation available notifications.

3.17.4 Outputs

- Notification delivery reports.

3.17.5 User Interface

- A notification center in the member's dashboard.

3.17.6 Performance

- Response time for reservation available notification delivery: < 2 seconds.
- Concurrent reservation available notification deliveries: Up to 20 per minute.

3.17.7 Security

- Secure communication channels for delivering notifications.
- Log reservation available notification activities for audit trail purposes.

3.18 Send Reservation Cancelled Notification

3.18.1 Description

- Notify members when their reserved book reservation is cancelled.

3.18.2 Inputs

- System-generated reservation cancelled events.
- Member ID.
- Notification content.

3.18.3 Processing

- Identify members with cancelled reservations.
- Trigger and send reservation cancelled notifications.

3.18.4 Outputs

- Notification delivery reports.

3.18.5 User Interface

- A notification center in the member's dashboard.

3.18.6 Performance

- Response time for reservation cancelled notification delivery: < 2 seconds.
- Concurrent reservation cancelled notification deliveries: Up to 15 per minute.

3.18.7 Security

- Secure communication channels for delivering notifications.
- Log reservation cancelled notification activities for audit trail purposes.

3.19 Non-Functional Requirements

3.19.1 Performance

3.19.1.1 Response Time

The system should ensure a response time of less than 3 seconds for user interactions, such as book searches, renewals, and profile updates.

3.19.1.2 Concurrent Users

The system should support a minimum of 200 concurrent users, ensuring optimal performance during peak hours.

3.20 Scalability

3.20.1 System Capacity

The system should be scalable to accommodate a growing database of up to 50,000 books and 5,000 members.

3.20.2 Performance Scaling

The system should be capable of dynamically scaling resources to handle increased user loads without compromising performance.

3.21 Reliability

3.21.1 System Availability

The system should have a minimum uptime of 99.9%, ensuring it is available to users at all times, except during scheduled maintenance.

3.21.2 Fault Tolerance

The system should be designed with fault-tolerant mechanisms to minimize downtime and ensure data integrity in case of hardware or software failures.

3.22 Security

3.22.1 Access Control

The system should implement role-based access control, restricting certain functionalities to authorized users (e.g., librarians having access to book management features).

3.22.2 Data Encryption

All sensitive data, including member information and transaction details, should be encrypted during transmission and storage to prevent unauthorized access.

3.22.3 Secure Transmission

The system should use secure communication protocols (e.g., HTTPS) to safeguard data during transmission over the network.

3.23 Usability

3.23.1 User Interface Design

The user interface should follow industry best practices for design, ensuring a user-friendly and intuitive experience for both members and librarians.

3.23.2 User Training

The system should provide easy-to-understand training materials or tutorials for new users to familiarize themselves with the functionalities.

3.24 Compatibility

3.24.1 Cross-Browser Compatibility

The system should be compatible with major web browsers such as Chrome, Firefox, Safari, and Edge.

3.24.2 Operating System Compatibility

The system should be accessible and fully functional on common operating systems including Windows, macOS, and Linux.

3.25 Data Integrity

3.25.1 Data Validation

The system should incorporate robust data validation mechanisms to ensure the accuracy and completeness of entered information.

3.25.2 Error Handling

Adequate error handling should be implemented to gracefully manage errors, preventing data corruption and ensuring a smooth user experience.

3.26 Compliance

3.26.1 Legal Compliance

The system should comply with all applicable laws and regulations related to library operations and data management.

3.26.2 Data Protection Laws

The system should adhere to data protection laws, ensuring the privacy and security of member information.

3.27 Audit Trail

3.27.1 Logging and Monitoring

The system should maintain detailed logs of user activities, providing an audit trail for security and compliance purposes.

3.28 Accessibility

3.28.1 Compliance with Accessibility Standards

The system should comply with accessibility standards (e.g., WCAG) to ensure it is accessible to users with disabilities.

3.29 External Interface Requirements

3.29.1 User Interfaces

3.29.1.1 Member Interface

The member interface should be accessible through standard web browsers and mobile devices, providing a consistent user experience.

3.29.1.2 Librarian Interface

The librarian interface should be optimized for desktop use, offering advanced functionalities for book management.

3.30 Hardware Interfaces

3.30.1 System Requirements

The system should be compatible with standard hardware configurations, requiring minimal resource specifications for optimal performance.

3.30.2 Barcode Scanners

The system should support common barcode scanners for efficient book management by librarians.

3.31 Software Interfaces

3.31.1 Database Management System

The system should be compatible with a widely used and reliable database management system, ensuring efficient data storage and retrieval.

3.31.2 External APIs

The system may integrate with external APIs for additional functionalities or data sources, following industry standards for API communication.

3.32 Communication Interfaces

3.32.1 Notifications

The system should utilize secure communication channels for sending notifications to all.

USE CASES:

User Use Cases:

Login:

Enables users (students and staff) to log into the library system.

Forget Password:

Allows users to recover or reset their forgotten passwords.

Logout:

Enables users to log out of the library system.

Registration:

Allows users (students and staff) to register for a library account.

Search Book:

Enables users to search for books in the library catalog.

Borrow Book:

Allows users to borrow books from the library.

Renew Book:

Permits users to renew borrowed books.

Return Book:

Allows users to return borrowed books to the library.

Profile Management:

Permits users to manage their profile information.

Feedback:

Allows users to provide feedback on library services.

Book History:

Allows users to view their borrowing history.

Book Fine:

Provides information about fines related to overdue books.

Librarian Use Cases:

Cancel Membership:

Allows librarians to cancel a user's library membership.

Fine:

Allows librarians to manage fines for overdue books.

Notification:

Enables librarians to send notifications to users.

Report:

Allows librarians to generate reports on library statistics.

Review Feedback:

Permits librarians to review user feedback.

Maintain Book Records:

Allows librarians to manage the library's book inventory.

Update Fine:

Permits librarians to update fine details for specific users.

USE CASES TABLE

INITIATOR	INITIATOR'S GOAL	PARTICIPANTS	USE CASE NAME
Student, Staff	Register for a library account	User	Registration
Student, Staff	Search for books in the library catalog	User	Search Book
Student, Staff	Borrow books from the library	User	Borrow Book
Student, Staff	Renew borrowed books	User	Renew Book
Student, Staff	Return borrowed books	User	Return Book
Student, Staff	Manage profile information	User	Profile Management
Student, Staff	Provide feedback on library services	User	Feedback
Student, Staff	Log into the library system	User	Login
Student, Staff	View borrowing history	User	Book History
Student, Staff	View fines related to overdue books	User	Book Fine
Student, Staff	Recover or reset forgotten passwords	User	Forget Password
Student, Staff	Log out of the library system	User	Logout
Librarian	Cancel a user's library membership	Librarian	Cancel Membership
Librarian	Manage fines for overdue books	Librarian	Fine
Librarian	Send notifications to users	Librarian	Notification
Librarian	Generate reports on library statistics	Librarian	Report
Librarian	Review user feedback	Librarian	Review Feedback
Librarian	Manage the library's book inventory	Librarian	Maintain Book Records
Librarian	Update fine details for specific users	Librarian	Update Fine
Database System	Send notifications for overdue books	Database System	Send Overdue Notification
Database System	Send notifications when reserved books are available	Database System	Send Reservation Available Notification
Database System	Send notifications when a reservation is canceled	Database System	Send Reservation Cancellation Notification

UC Description:

UC-1	**LOGIN**
Related Requirements	FR1
Initiating Actors	Student, Staff
Actor's Goal	To log into the library system.
Participating Actors	System
Preconditions	User is registered in the system.
Postconditions	User is successfully logged into the system.
Flow of Events (Main Success Scenario)	- User enters a unique username and password. - System authenticates the user. - Upon successful authentication, the user gains access to personalized features.
Flow of Events (Alternate Scenario)	- If the entered username or password is incorrect, the system prompts the user to re-enter.

UC-2	**FORGET PASSWORD**
Related Requirements	FR2
Initiating Actors	Student, Staff
Actor's Goal	To recover or reset a forgotten password.
Participating Actors	System
Preconditions	User forgets their password.
Postconditions	User successfully resets the password.
Flow of Events (Main Success Scenario)	- User selects the "Forgot Password" option. - User provides necessary information for identity verification. - System verifies the information and allows the user to reset the password.
Flow of Events (Alternate Scenario)	- If the provided information for identity verification is incorrect, the system informs the user to provide correct details.

UC-3	**LOGOUT**
Related Requirements	FR3
Initiating Actors	Student, Staff
Actor's Goal	To log out of the library system.
Participating Actors	System
Preconditions	User is logged into the system.
Postconditions	User is successfully logged out.
Flow of Events (Main Success Scenario)	- User selects the "Logout" option. - System logs the user out, terminating the session.

UC-4 REGISTRATION

Related Requirements	FR4
Initiating Actors	Student, Staff
Actor’s Goal	To register for a library account.
Participating Actors	System
Preconditions	User is not registered in the system.
Postconditions	User is successfully registered and receives confirmation.
Flow of Events (Main Success Scenario)	- User selects the registration option. - User provides necessary information (name, contact details). - System validates and creates a new user account. - User receives confirmation.
Flow of Events (Alternate Scenario)	- If the provided information is incomplete or invalid, the system prompts the user to correct them. - If the email is already registered, the system informs the user to use a different email.

UC-5 SEARCH BOOK

Related Requirements	FR5
Initiating Actors	Student, Staff
Actor’s Goal	To search for books in the library catalog.
Participating Actors	System
Preconditions	User is logged into the system.
Postconditions	System displays a list of matching books.
Flow of Events (Main Success Scenario)	- User enters book details (title, author, etc.) into the search bar. - System displays a list of matching books. - User can apply filters to refine the search results.

UC-6 BORROW BOOK

Related Requirements	FR6
Initiating Actors	Student, Staff
Actor’s Goal	To borrow books from the library.
Participating Actors	System
Preconditions	User is logged into the system and has not reached the maximum borrowing limit.
Postconditions	System updates the book status, and a due date is assigned.
Flow of Events (Main Success Scenario)	- User selects a book for borrowing. - System updates the book status to "checked out" for the user. - Due date is assigned for the borrowed book.

UC-7 RENEW BOOK

Related Requirements	FR7
Initiating Actors	Student, Staff
Actor’s Goal	To renew borrowed books.
Participating Actors	System
Preconditions	User is logged into the system and has borrowed books.
Postconditions	Due date for the book is extended.
Flow of Events (Main Success Scenario)	- User selects the option to renew a book. - System validates the renewal request. - Due date for the book is extended.

UC-8 RETURN BOOK

Related Requirements	FR8
Initiating Actors	Student, Staff
Actor's Goal	To return borrowed books to the library.
Participating Actors	System
Preconditions	User is logged into the system and has borrowed books.
Postconditions	System updates the book status to "available," and fines are calculated if overdue.
Flow of Events (Main Success Scenario)	- User selects the option to return a book. - User scans the book's barcode. - System updates the book status to "available." - If the book is returned overdue, the system calculates fines.

UC-9 PROFILE MANAGEMENT

Related Requirements	FR9
Initiating Actors	Student, Staff
Actor's Goal	To manage their profile information.
Participating Actors	System
Preconditions	User is logged into the system.
Postconditions	User's profile information is successfully updated.
Flow of Events (Main Success Scenario)	- User selects the option to manage their profile. - User updates contact information or changes the password.

UC-10 FEEDBACK

Related Requirements	FR10
Initiating Actors	Student, Staff
Actor's Goal	To provide feedback on library services.
Participating Actors	System
Preconditions	User is logged into the system.
Postconditions	System records the feedback.
Flow of Events (Main Success Scenario)	- User selects the option to provide feedback. - User enters feedback comments. - System records the feedback.

UC-11 BOOK HISTORY

Related Requirements	FR11
Initiating Actors	Student, Staff
Actor's Goal	To view their borrowing history.
Participating Actors	System
Preconditions	User is logged into the system.
Postconditions	System displays a list of previously borrowed books.
Flow of Events (Main Success Scenario)	- User selects the option to view their book history. - System displays a list of previously borrowed books.

UC-12	**BOOK FINE**
Related Requirements	FR12
Initiating Actors	Student, Staff
Actor's Goal	To view information about fines related to overdue books.
Participating Actors	System
Preconditions	User is logged into the system.
Postconditions	System displays any fines associated with overdue books.
Flow of Events (Main Success Scenario)	- User selects the option to view fine details. - System displays any fines associated with overdue books.

UC-13	**CANCEL MEMBERSHIP**
Related Requirements	FR13
Initiating Actors	Librarian
Actor's Goal	To cancel a user's library membership.
Participating Actors	System
Preconditions	Librarian is logged into the system.
Postconditions	User's membership status is updated to "canceled."
Flow of Events (Main Success Scenario)	- Librarian selects the option to cancel membership. - Librarian provides the user ID for cancellation. - System updates the user status to "canceled."

UC-14	**FINE**
Related Requirements	FR14
Initiating Actors	Librarian
Actor's Goal	To manage fines for overdue books.
Participating Actors	System
Preconditions	Librarian is logged into the system.
Postconditions	Librarian views and updates fine records.
Flow of Events (Main Success Scenario)	- Librarian selects the option to manage fines. - Librarian views and updates fine records for overdue books. - Librarian can view fine history.

UC-15	**NOTIFICATION**
Related Requirements	FR15
Initiating Actors	Librarian
Actor's Goal	To send notifications to users.
Participating Actors	System
Preconditions	Librarian is logged into the system.
Postconditions	Notifications are sent to relevant users.
Flow of Events (Main Success Scenario)	- Librarian selects the option to send notifications. - Librarian chooses the type of notification (overdue, reservation available, cancellation). - System sends notifications to the relevant users.

UC-16	**REPORT**
Related Requirements	FR16
Initiating Actors	Librarian
Actor’s Goal	To generate reports on library statistics.
Participating Actors	System
Preconditions	Librarian is logged into the system.
Postconditions	System generates and presents the report to the librarian.
Flow of Events (Main Success Scenario)	- Librarian selects the option to generate reports. - Librarian chooses report parameters (e.g., circulation, popular genres). - System generates and presents the report to the librarian.

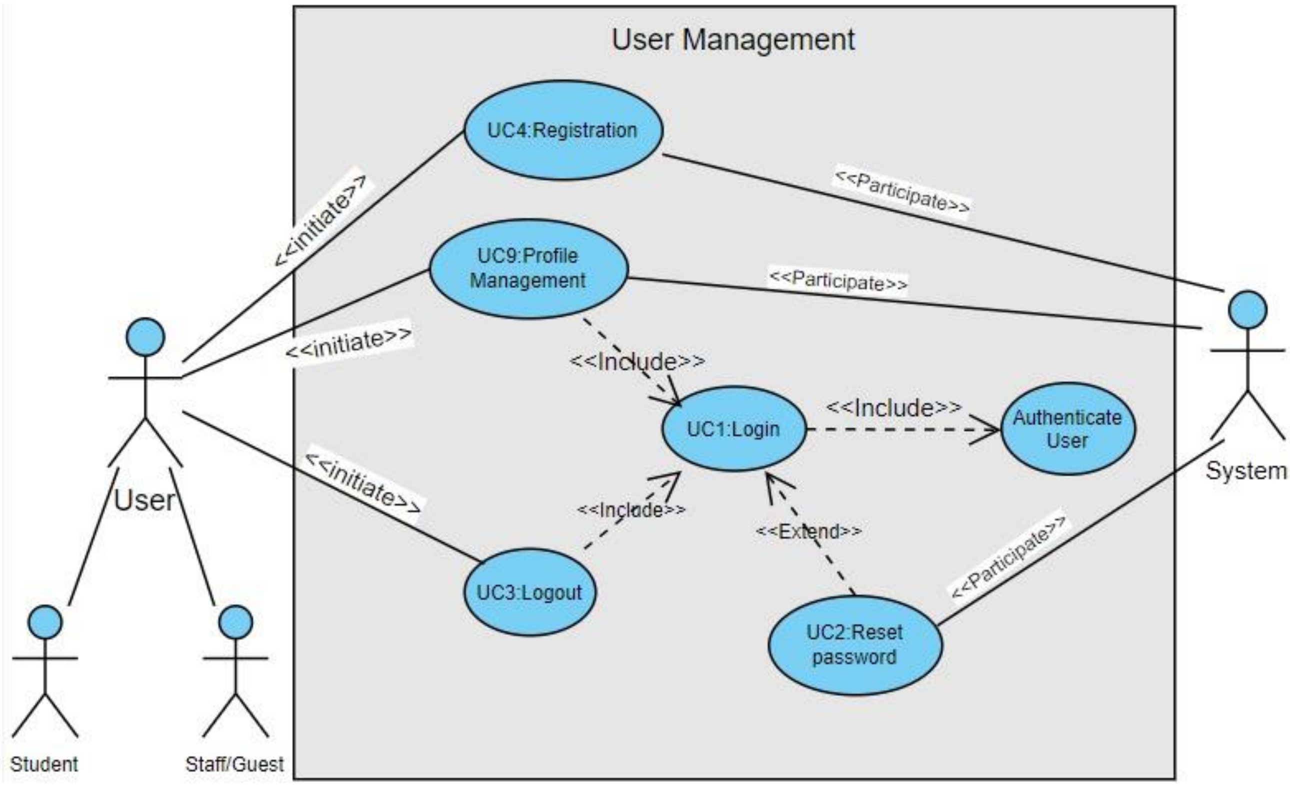
UC-17	**REVIEW FEEDBACK**
Related Requirements	FR17
Initiating Actors	Librarian
Actor’s Goal	To review user feedback.
Participating Actors	System
Preconditions	Librarian is logged into the system.
Postconditions	Librarian can view and respond to user feedback.
Flow of Events (Main Success Scenario)	- Librarian selects the option to review feedback. - Librarian can view and respond to user feedback.

UC-18	**MAINTAIN BOOK RECORDS**
Related Requirements	FR18
Initiating Actors	Librarian
Actor’s Goal	To manage the library's book inventory.
Participating Actors	System
Preconditions	Librarian is logged into the system.
Postconditions	Librarian can add, remove, or update book records.
Flow of Events (Main Success Scenario)	- Librarian selects the option to maintain book records. - Librarian can add new books to the catalog. - Librarian can remove books from the catalog. - Librarian can update existing book information.

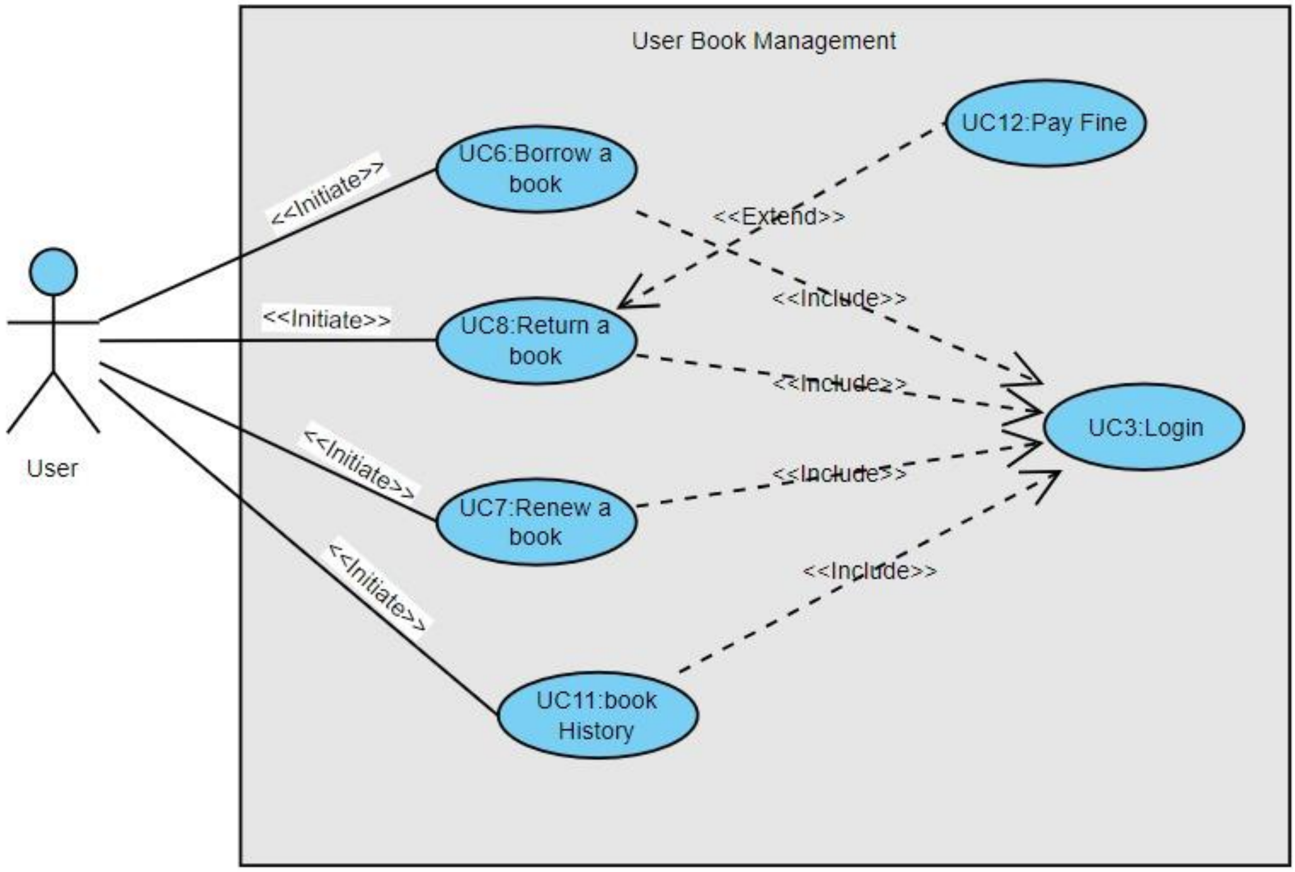
UC-19	**UPDATE FINE**
Related Requirements	FR19
Initiating Actors	Librarian
Actor’s Goal	To update fine details for specific users.
Participating Actors	System
Preconditions	Librarian is logged into the system.
Postconditions	Librarian updates fine details for specific users.
Flow of Events (Main Success Scenario)	- Librarian selects the option to update fines. - Librarian provides the user ID and updates fine information.

USE CASE DIAGRAMS:

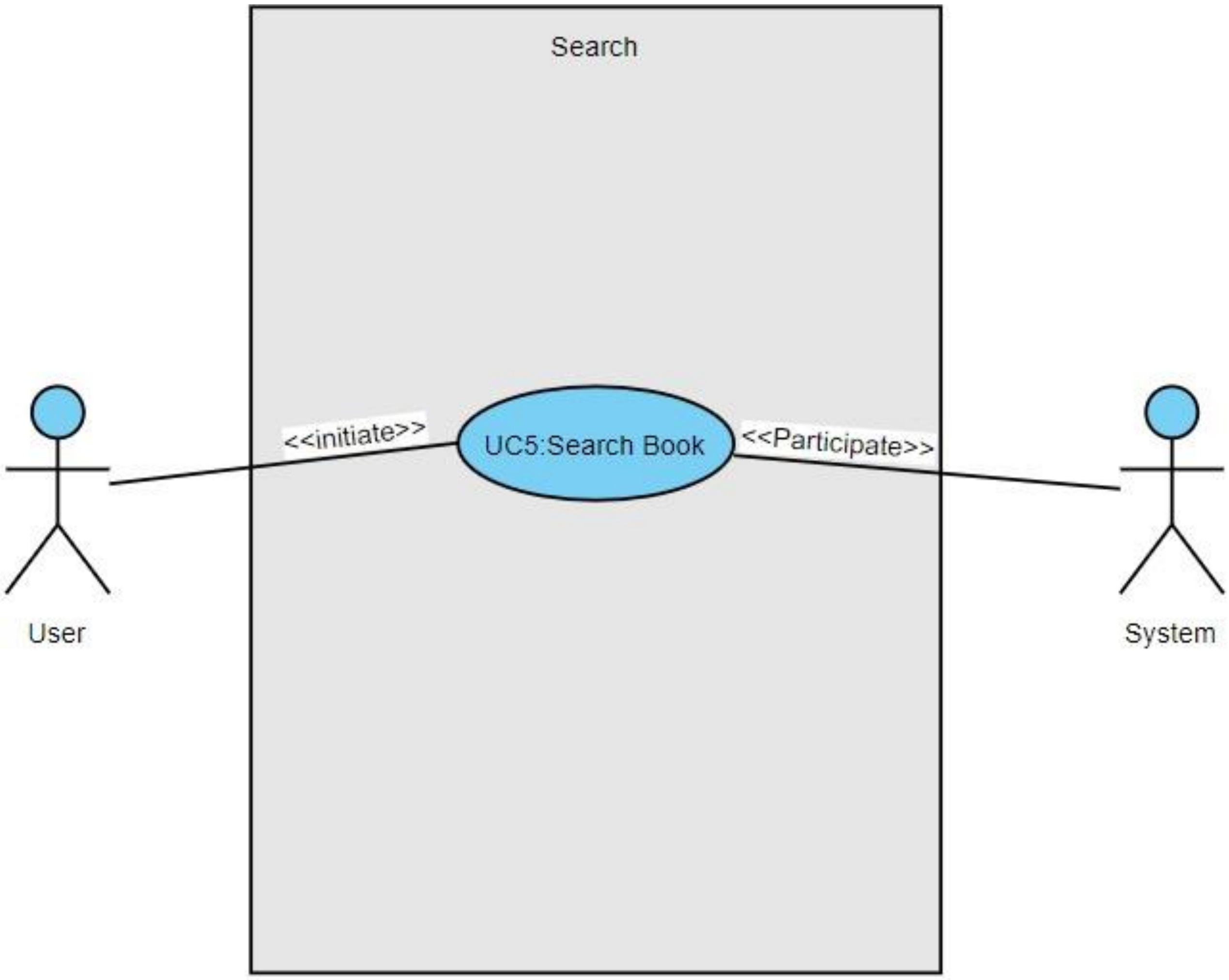
User Management :



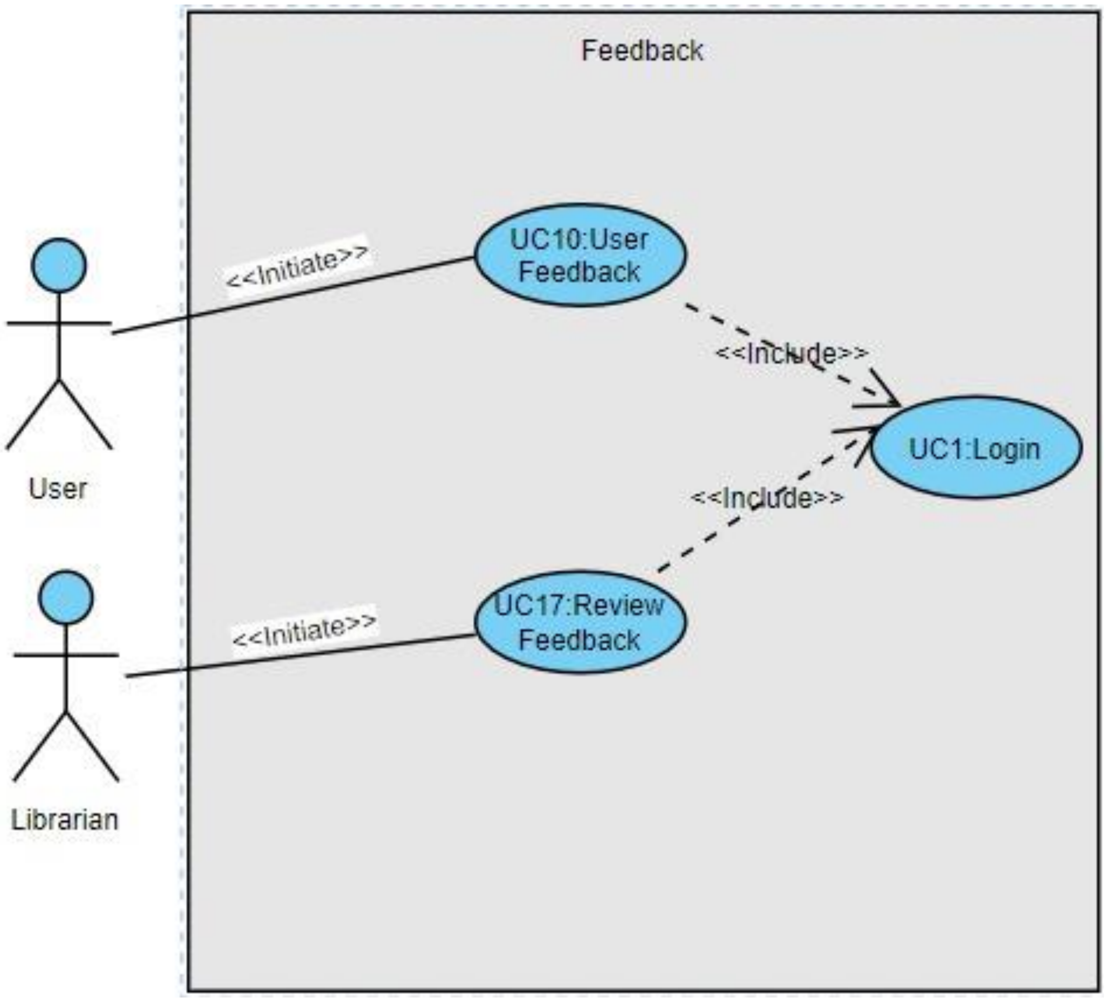
User Book Management:



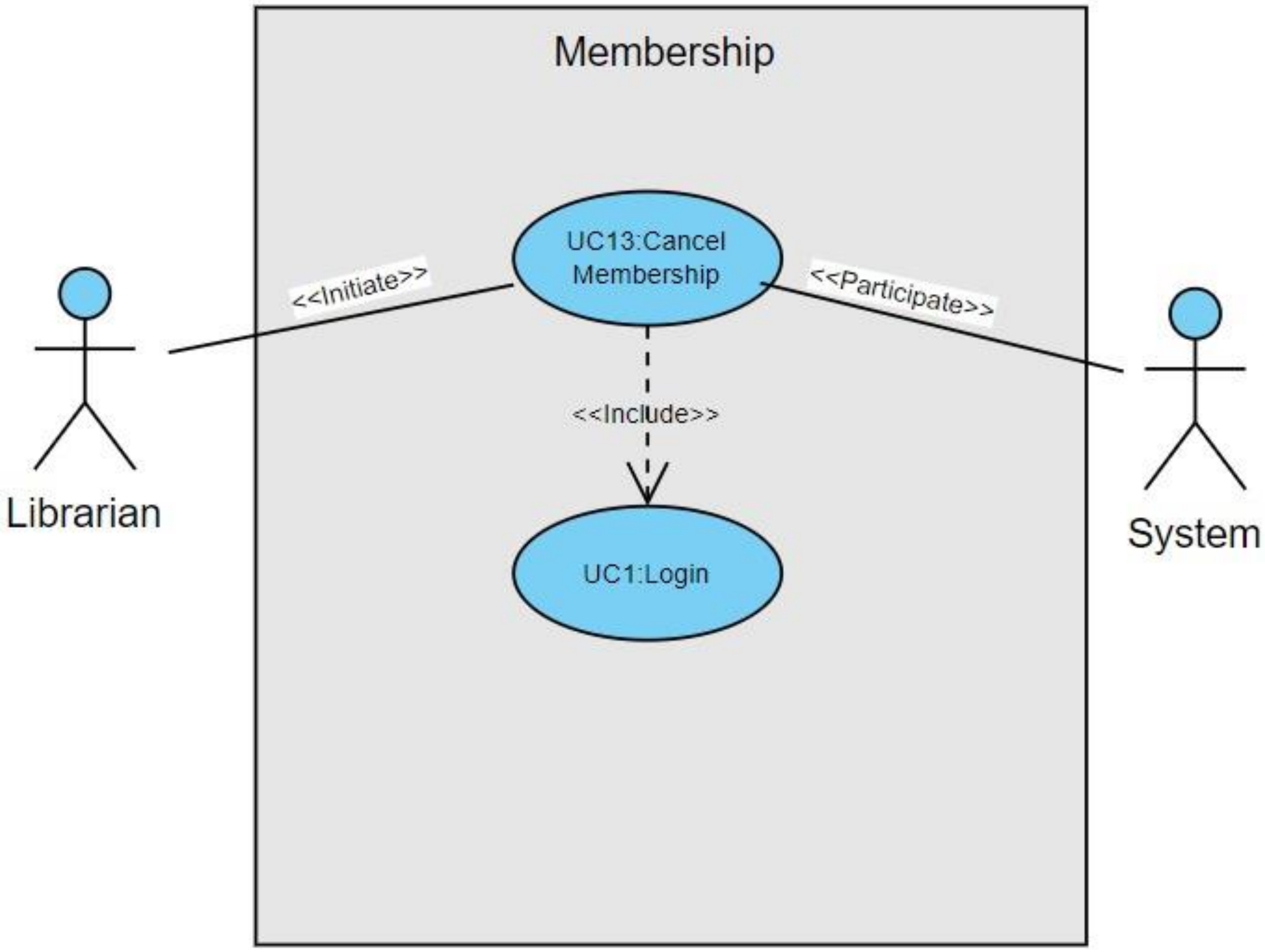
Book Searching:



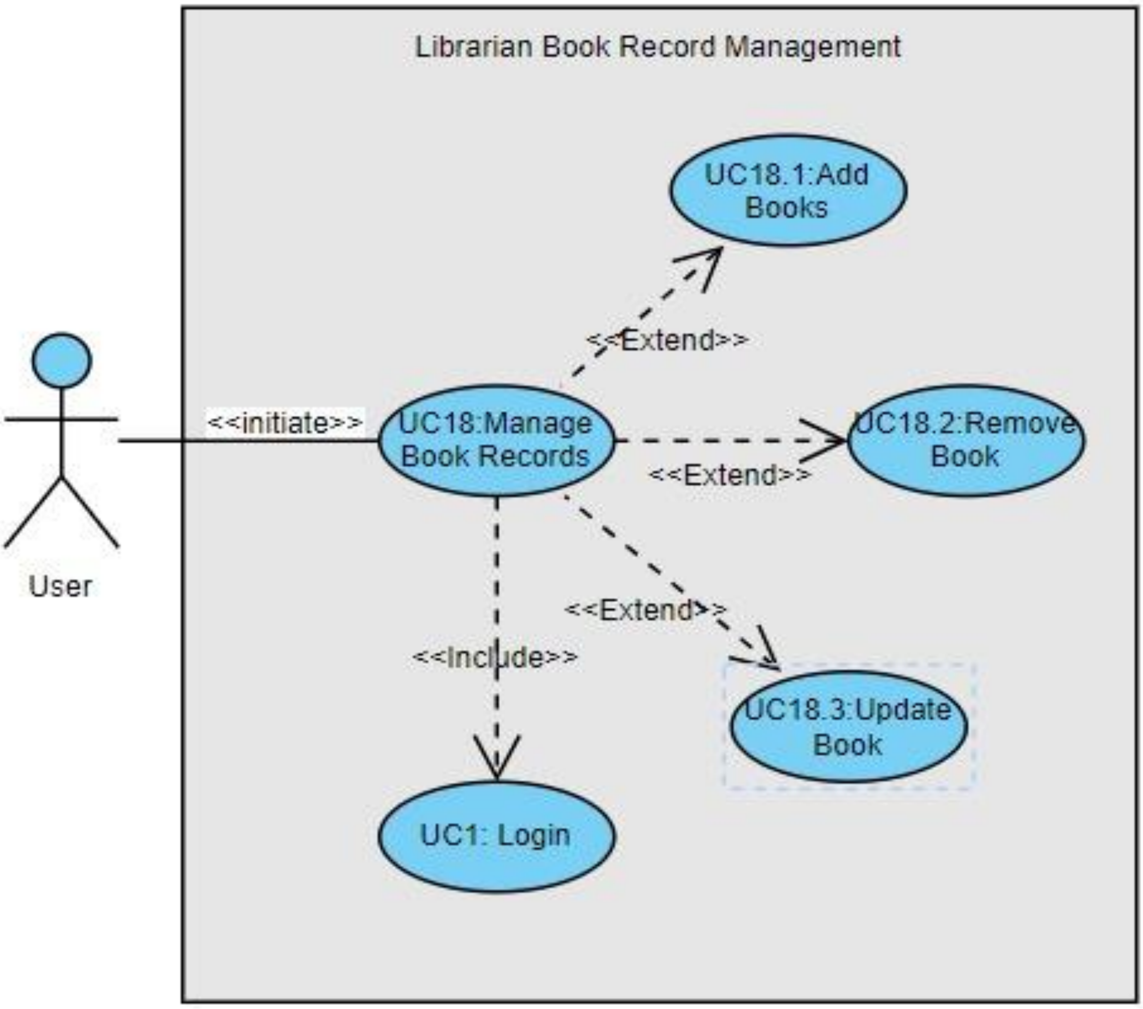
User Feedback and Review Feedback:



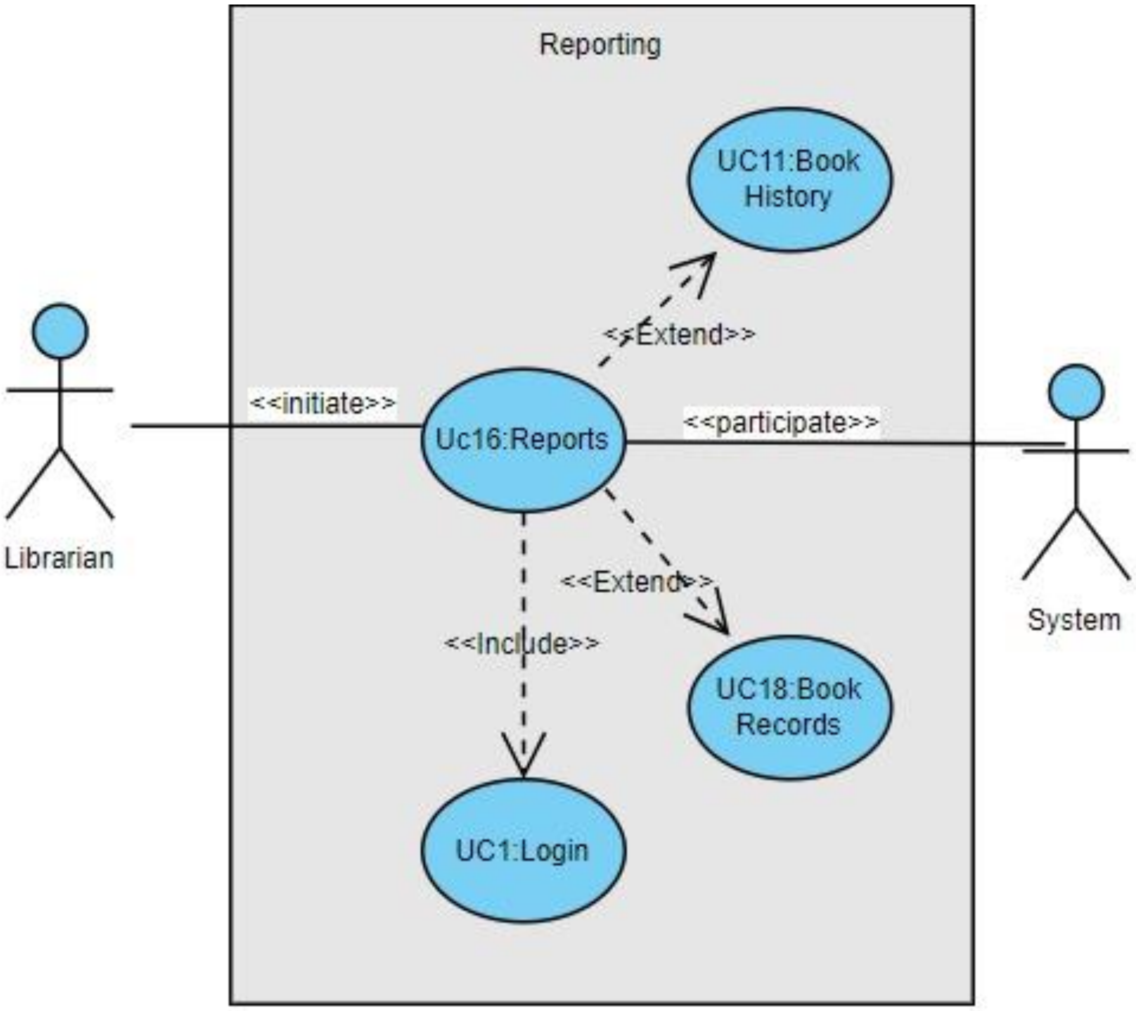
Membership:


Membership
UC13:Cancel Membership
<<Initiate>>
<<Participate>>
<<Include>>
UC1:Login
Librarian
System

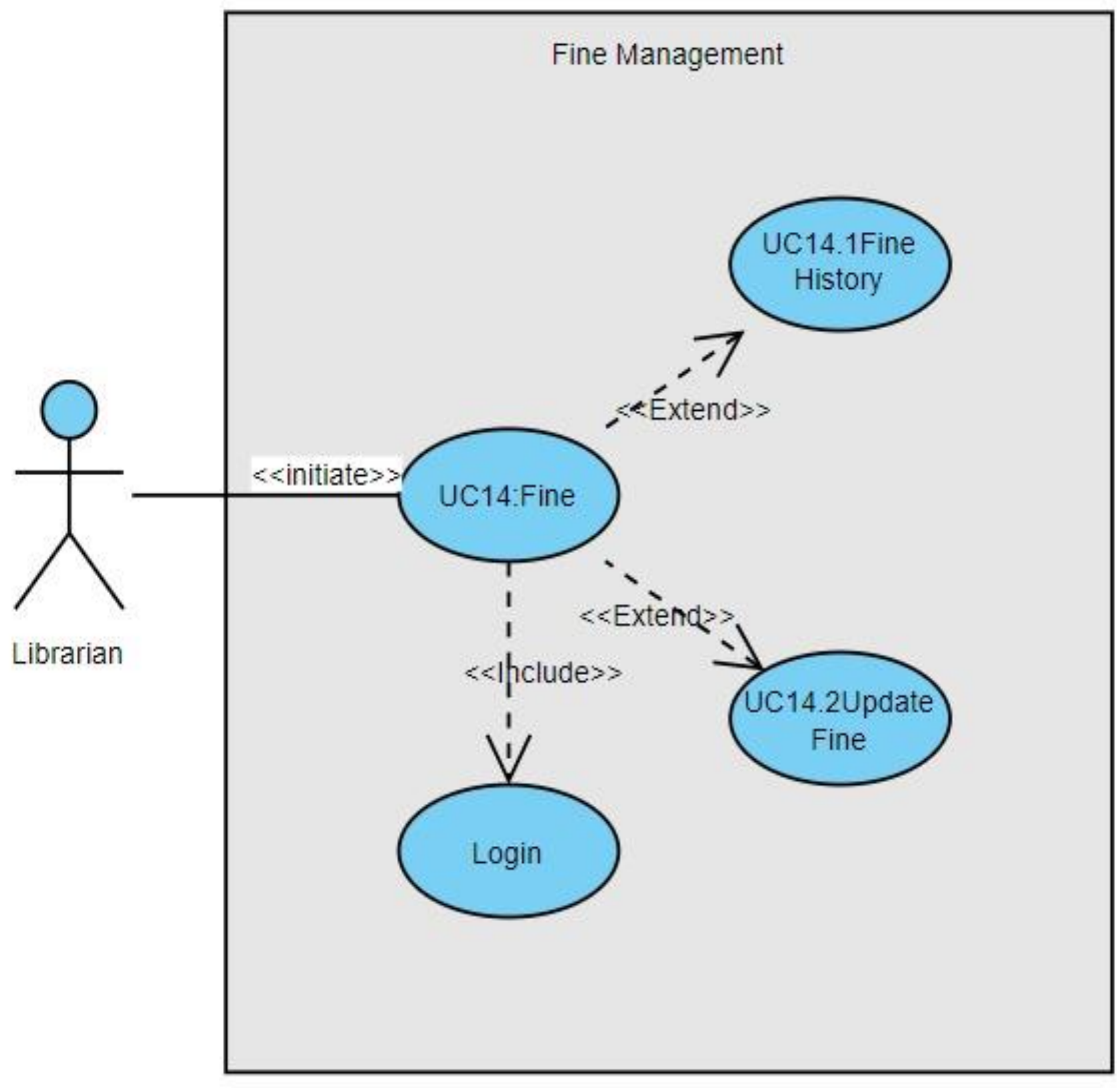
Librarian Book Record Management:



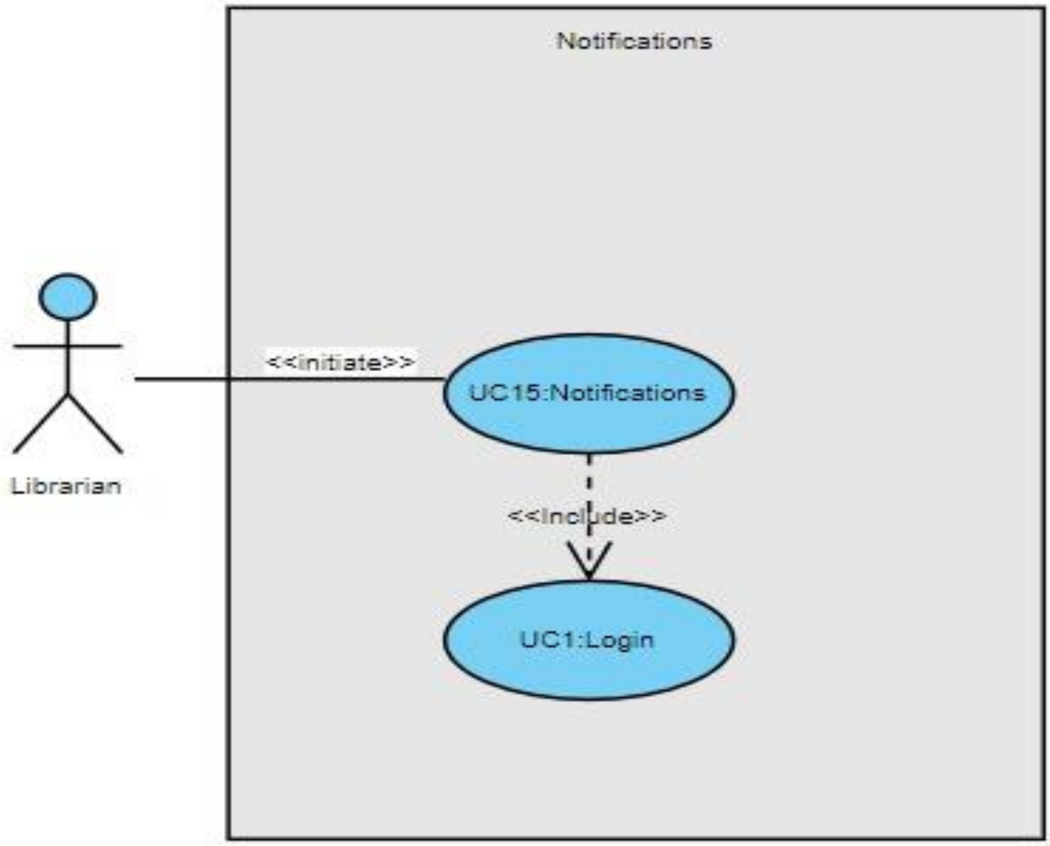
Reporting:


Reporting
UC11:Book History
<<Extend>>
<<initiate>>
Uc16:Reports
<<participate>>
Librarian
System
<<Extend>>
<<Include>>
UC18:Book Records
UC1:Login

Fine Management:

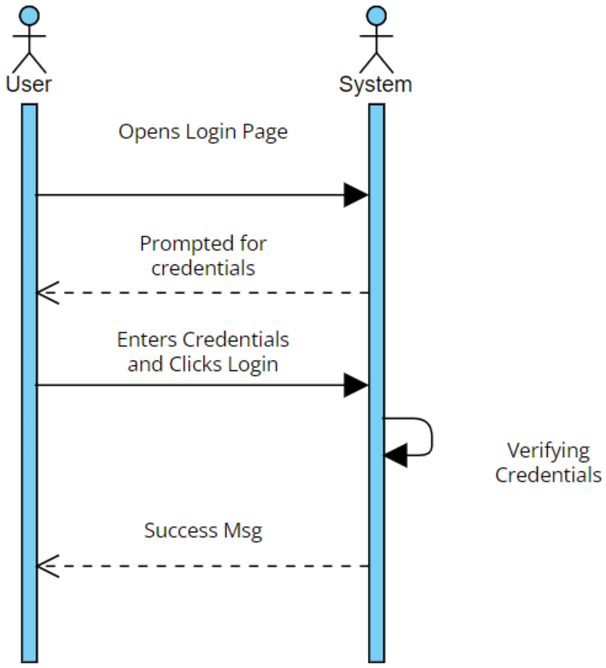


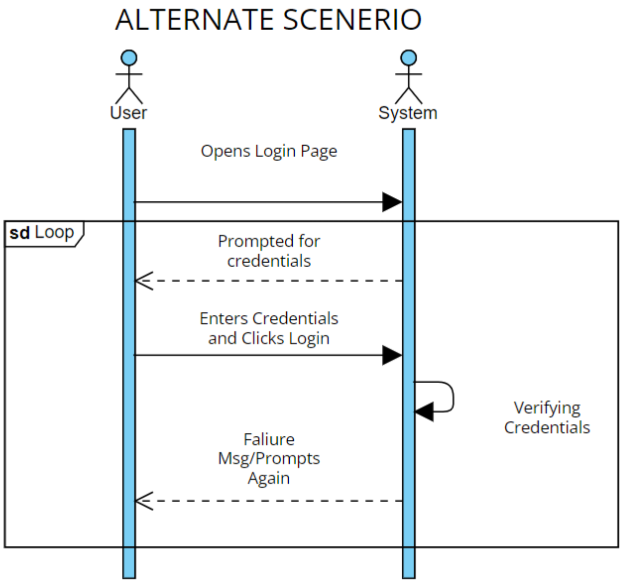
Notification:



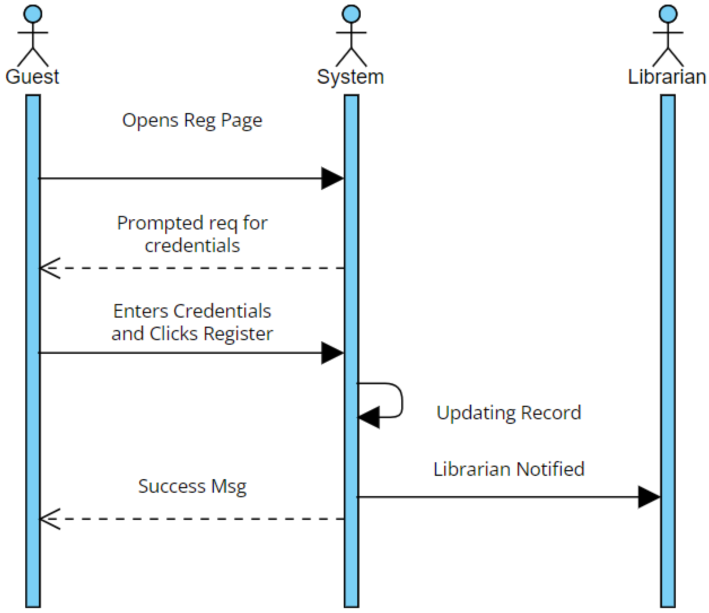
SYSTEM SEQUENCE DIAGRAMS:

. LOGIN:

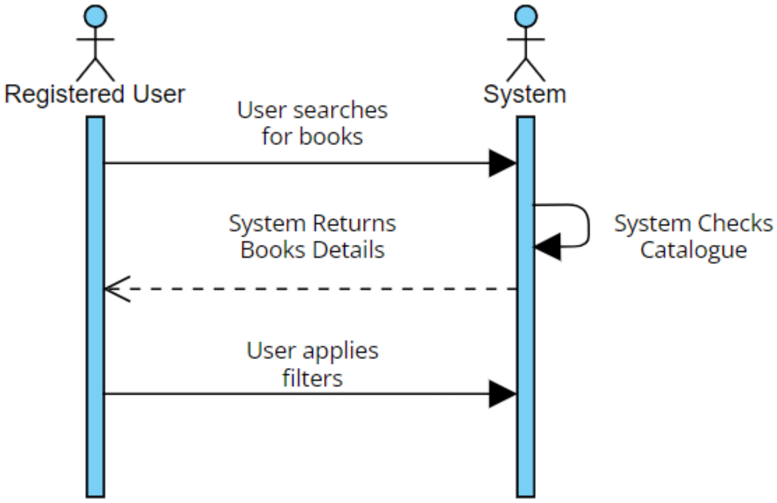




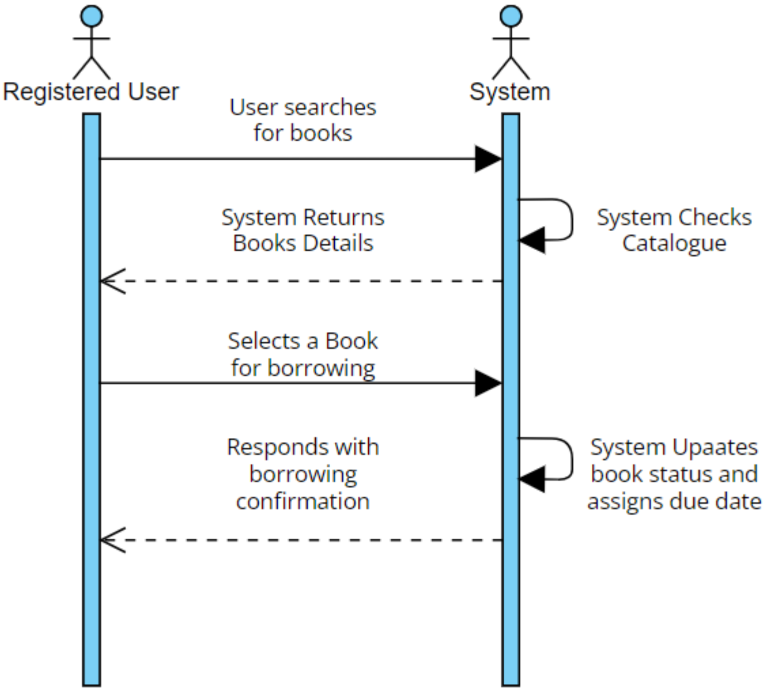
2. REGISTRATION:



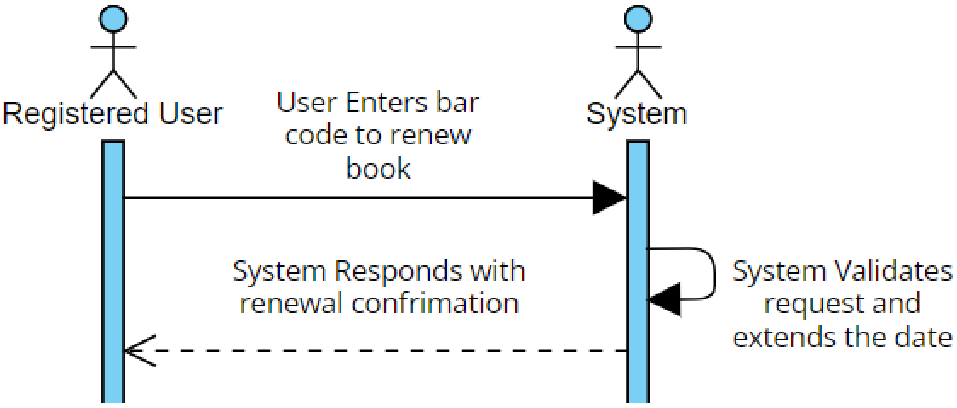
3. SEARCH BOOK:



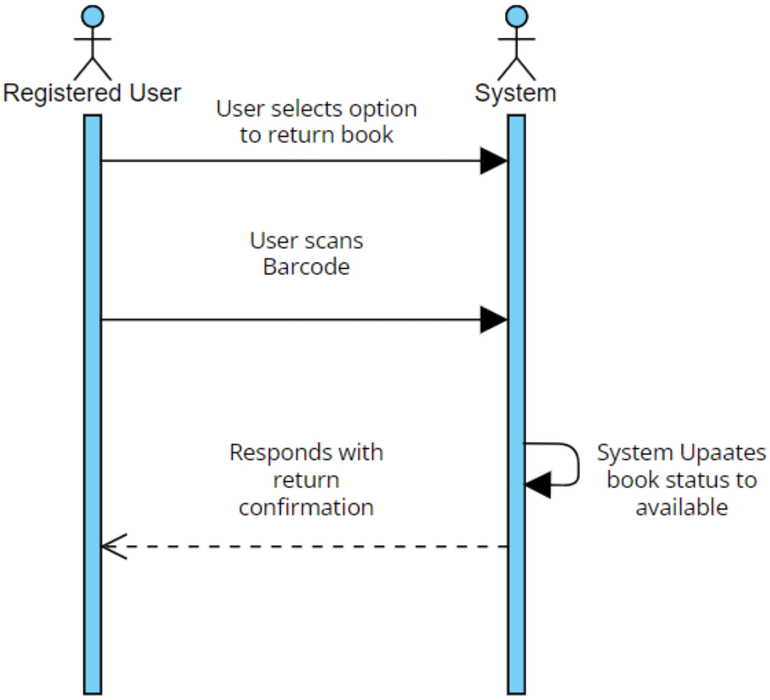
4. BORROW BOOK:


Registered User
System
User searches for books
System Returns Books Details
System Checks Catalogue
Selects a Book for borrowing
Responds with borrowing confirmation
System Upaates book status and assigns due date

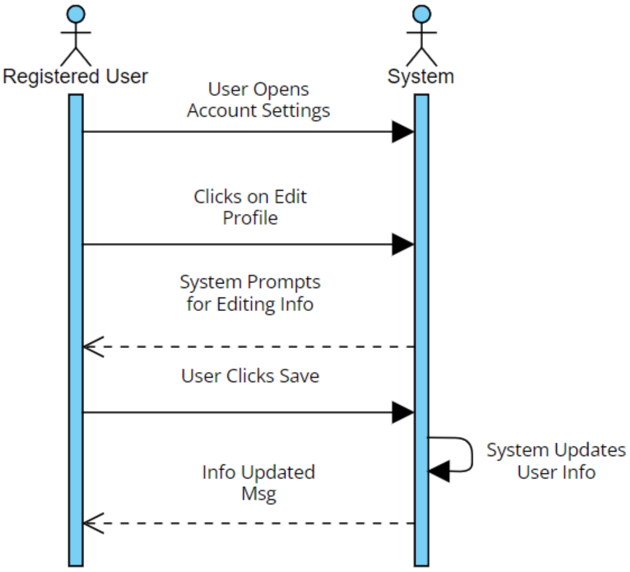
5. RENEW BOOK:


Registered User
System
User Enters bar code to renew book
System Responds with renewal confrimation
System Validates request and extends the date

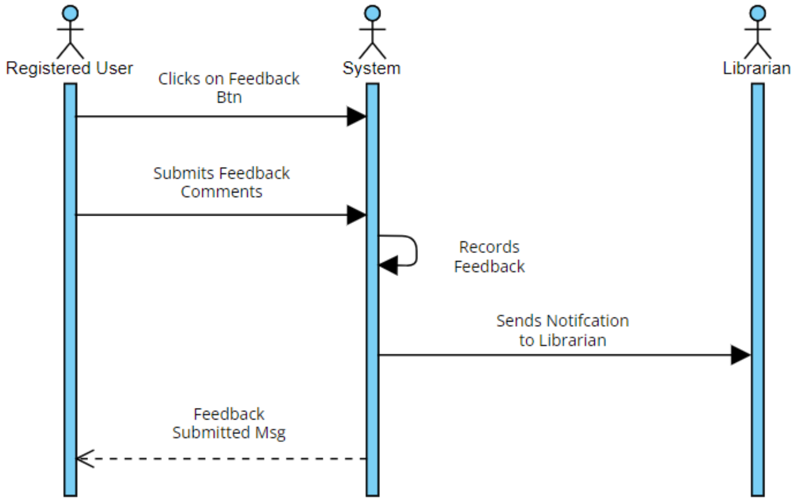
6. RETURN BOOK:


Registered User
System
User selects option to return book
User scans Barcode
Responds with return confirmation
System Upaates book status to available

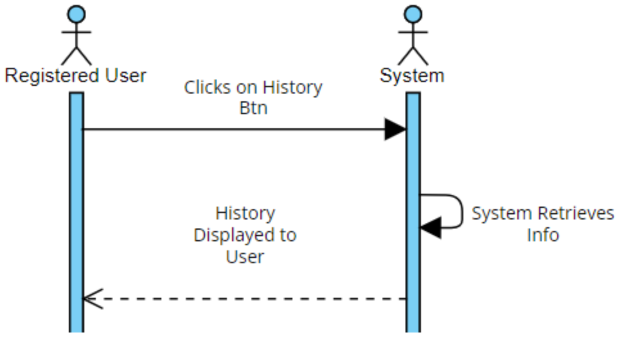
7. PROFILE MANAGEMENT:



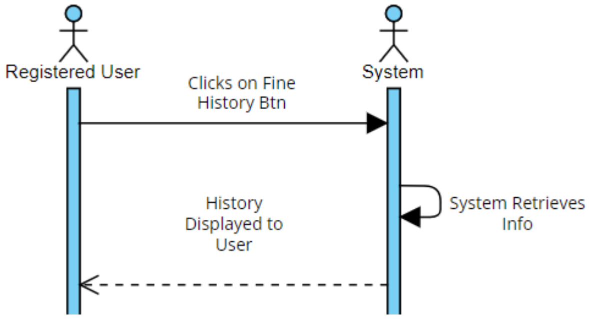
8. FEEDBACK:



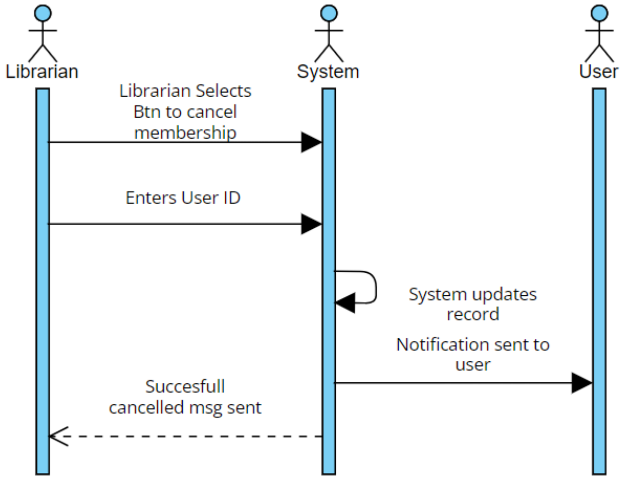
9. BOOK HISTORY:



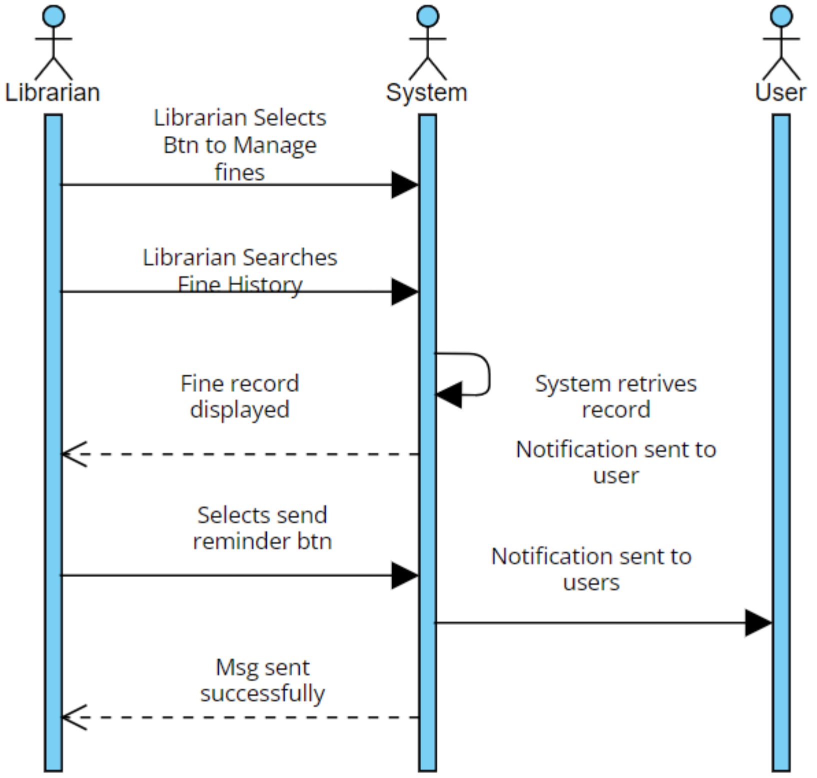
10. BOOK FINE:



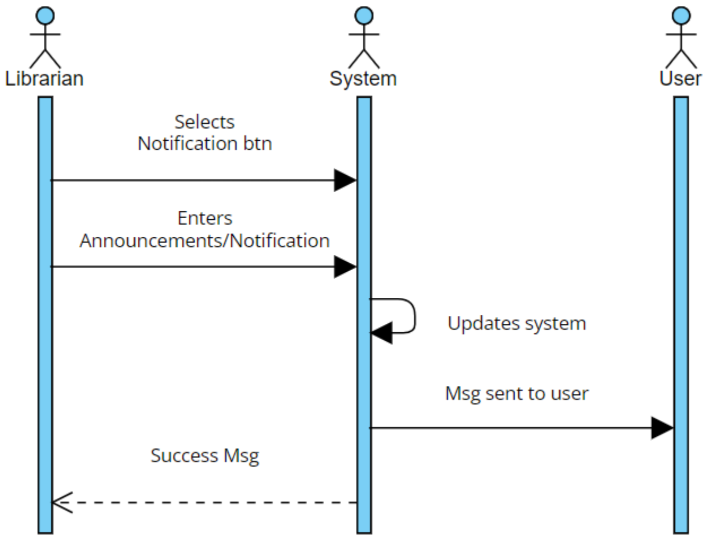
11. CANCEL MEMBERSHIP:


Librarian
System
User
Librarian Selects Btn to cancel membership
Enters User ID
System updates record
Notification sent to user
Succesfull cancelled msg sent

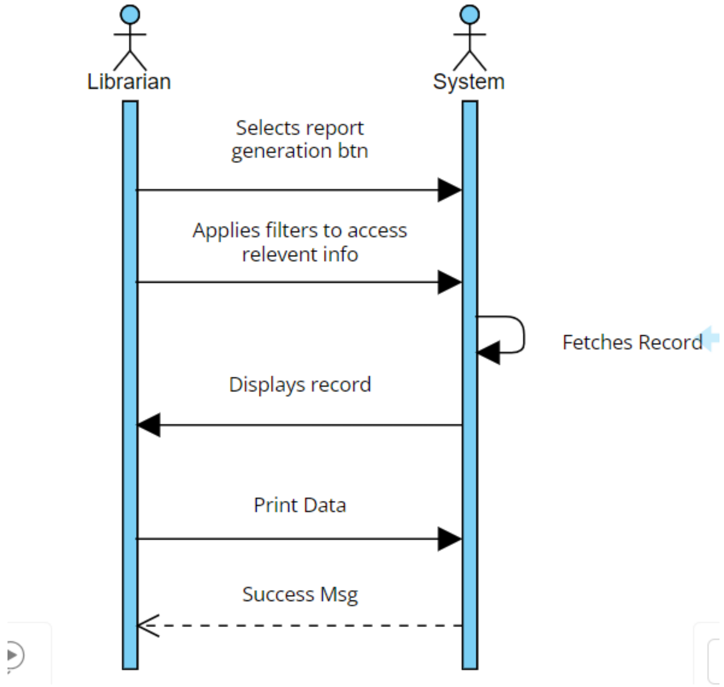
12. FINE:


Librarian
System
User
Librarian Selects Btn to Manage fines
Librarian Searches Fine History
Fine record displayed
System retrives record
Notification sent to user
Selects send reminder btn
Notification sent to users
Msg sent successfully

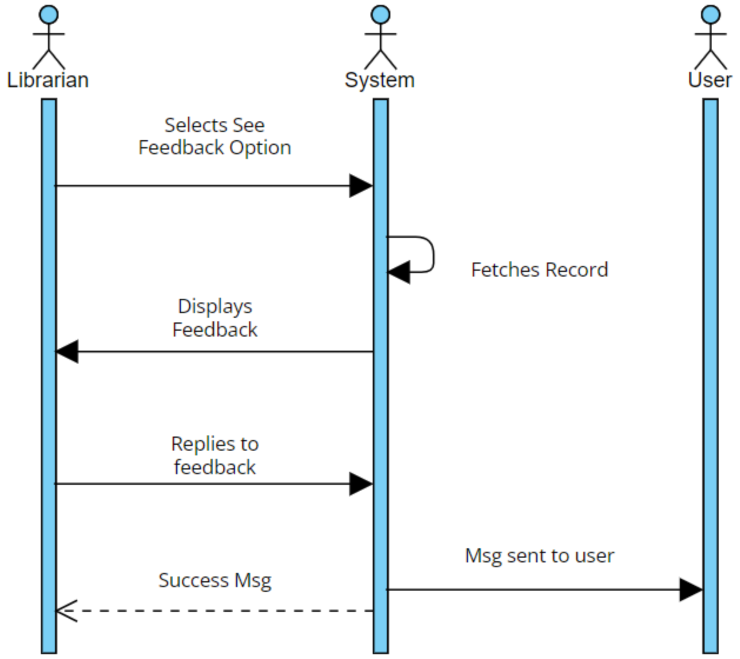
13. Notification/ Announcements:



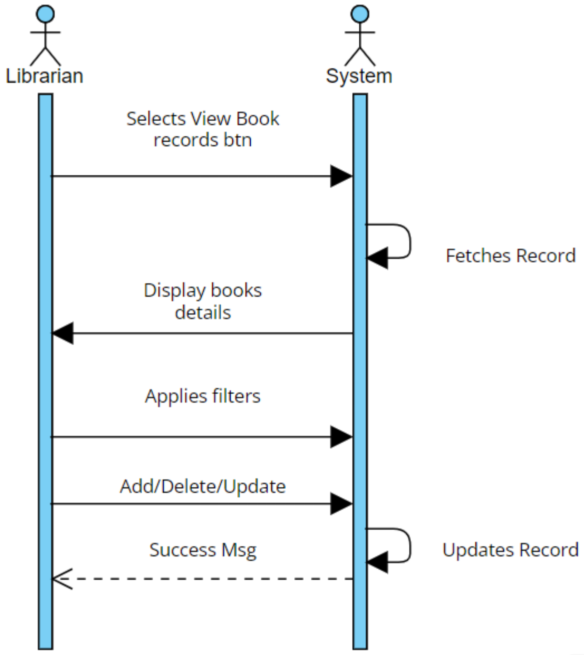
14. REPORTS:


Librarian
System
Selects report generation btn
Applies filters to access relevent info
Fetches Record
Displays record
Print Data
Success Msg

15. REVIEW FEEDBACK:

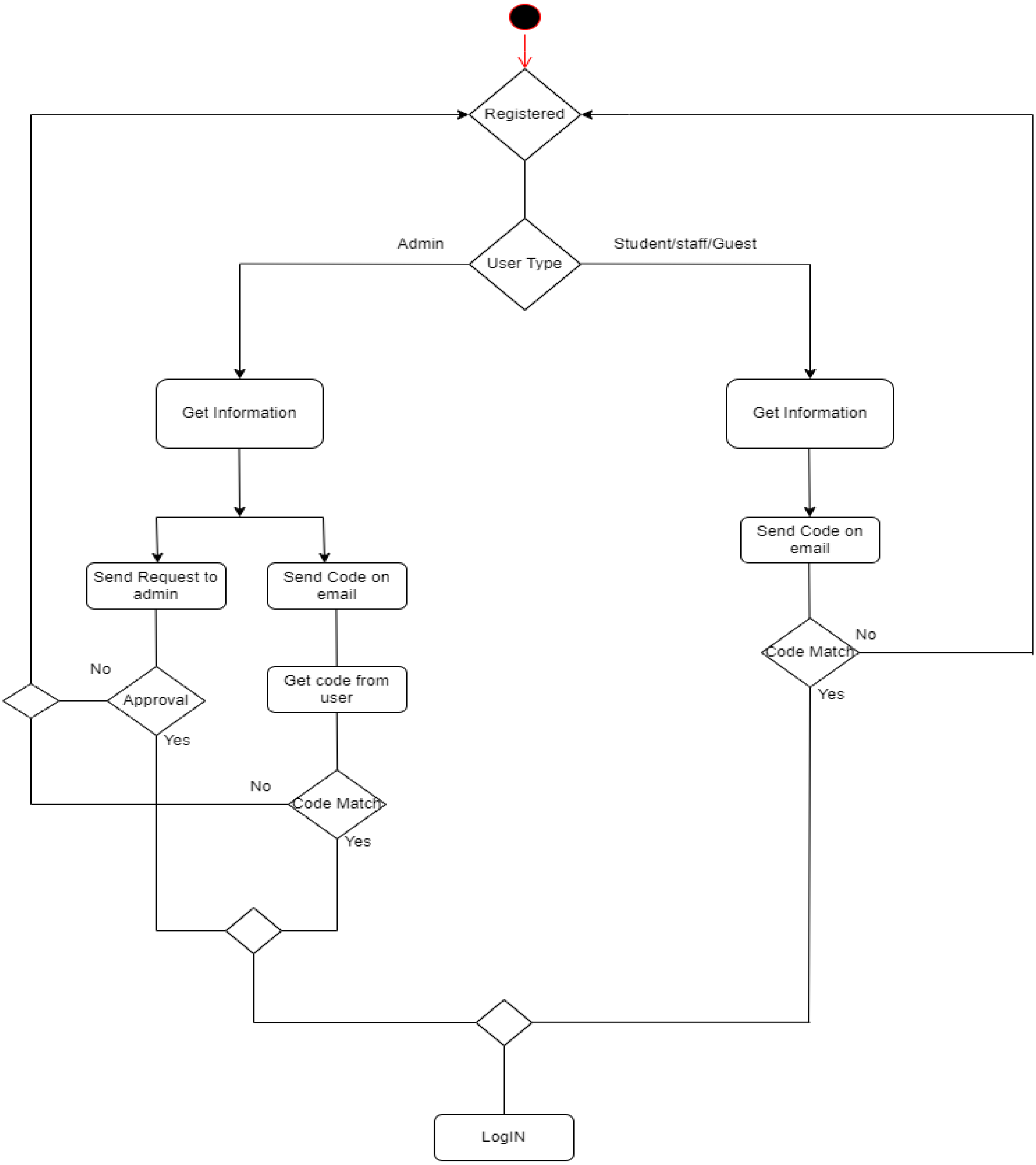


16. MAINTAIN BOOKS RECORDS:



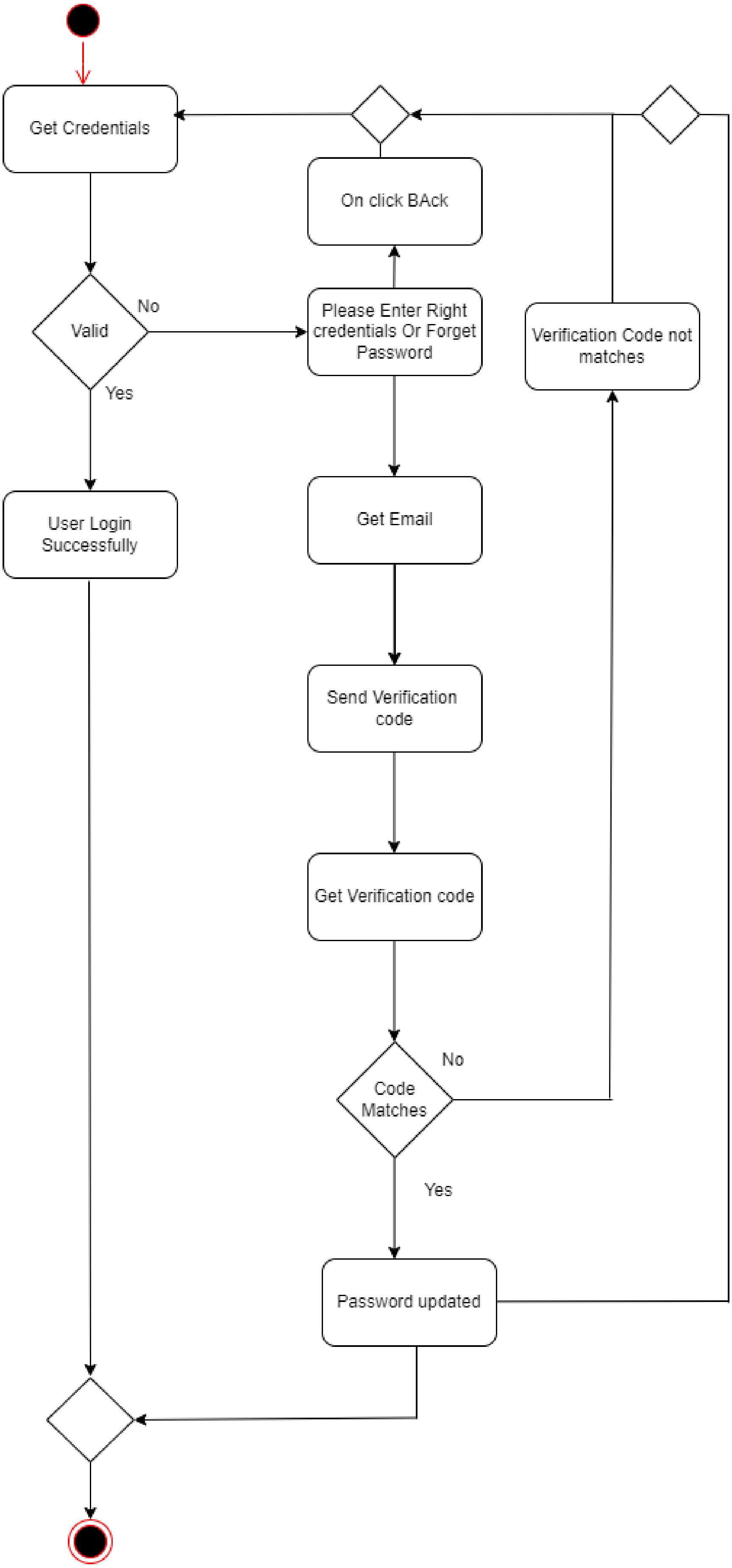
ACTIVITY DIAGRAMS:

Registration



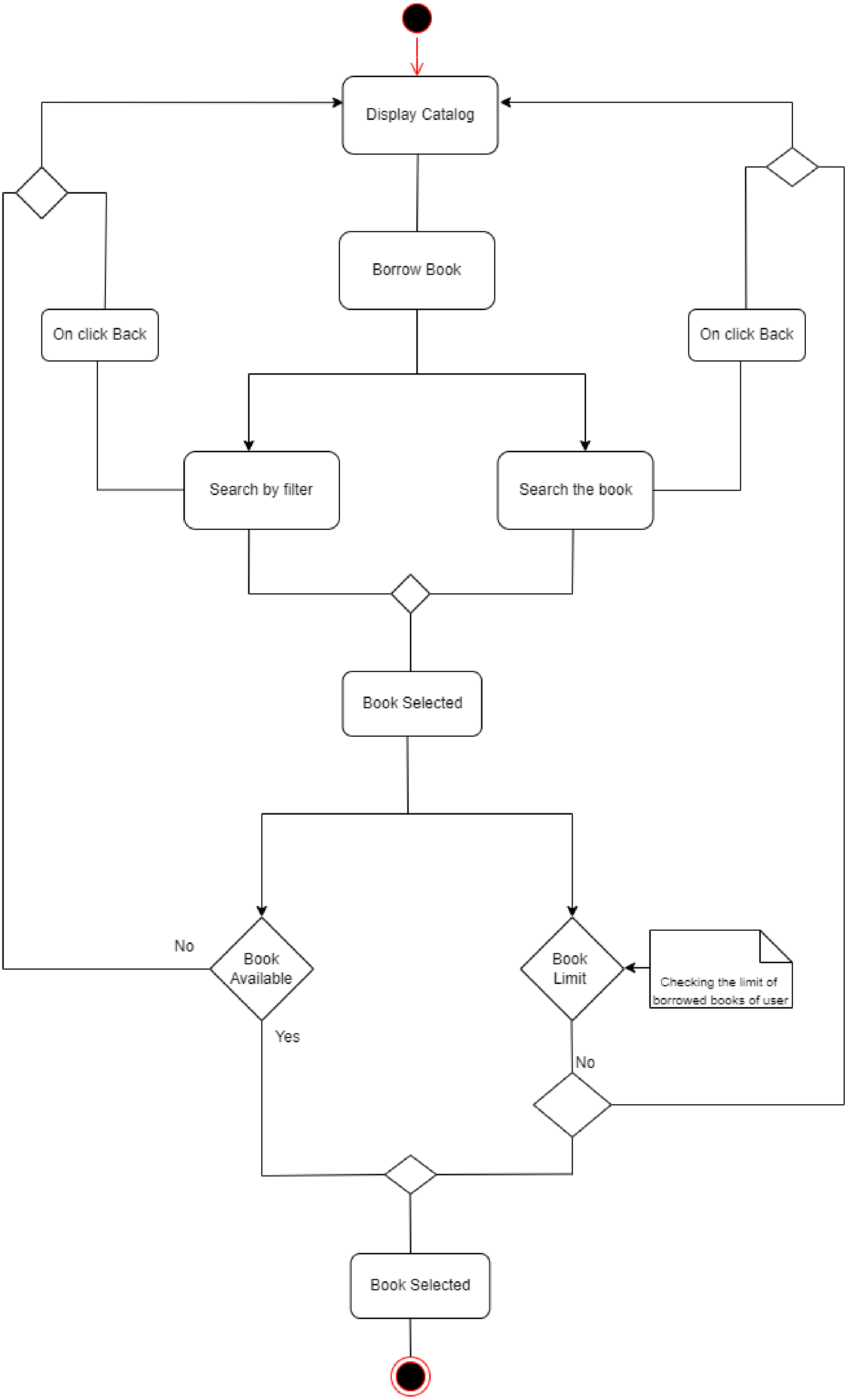
LibraSys (LMS)

LOGIN



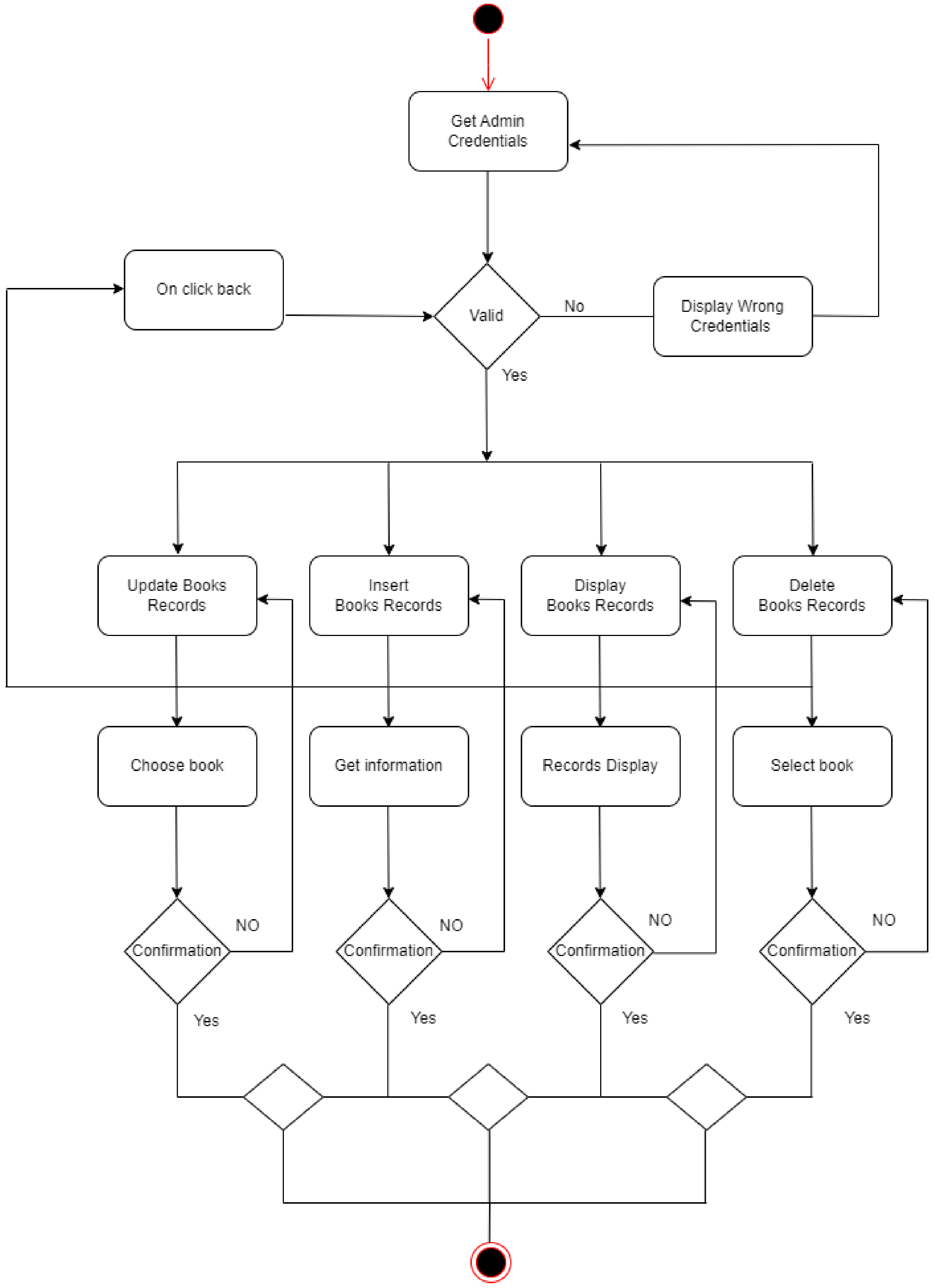
LibraSys (LMS)

Borrow Book

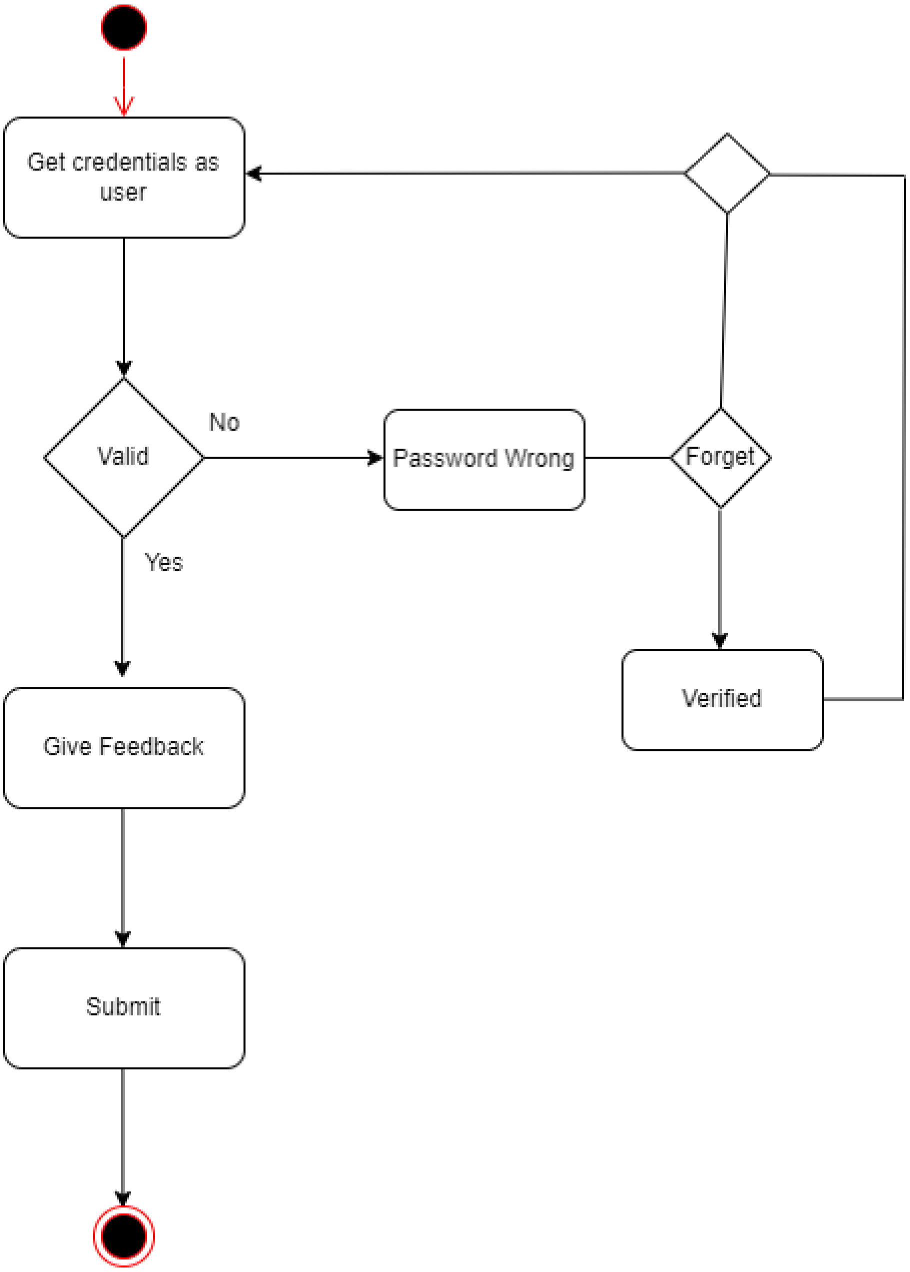


LibraSys (LMS)

Maintain Records

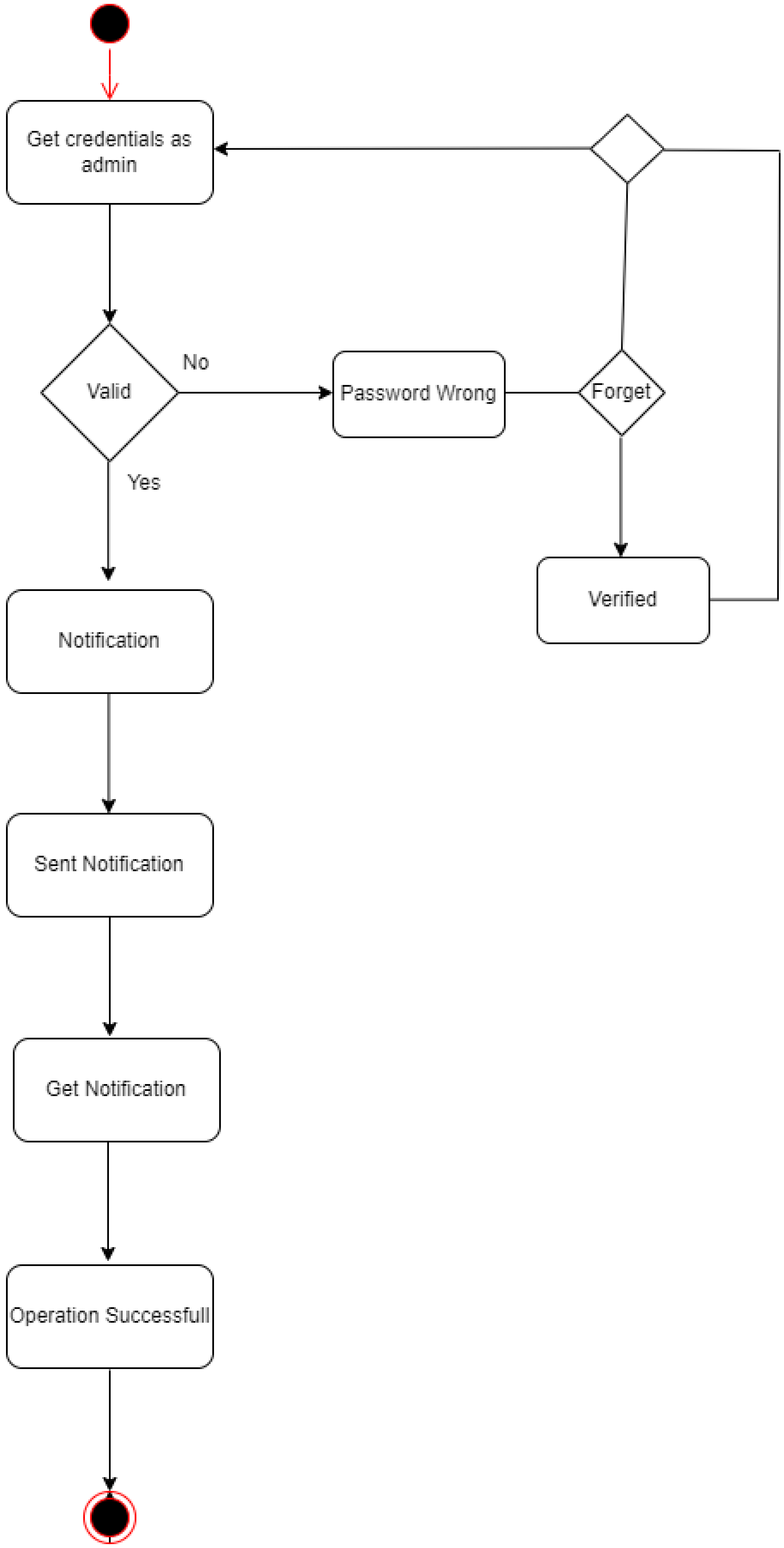


User Feedback

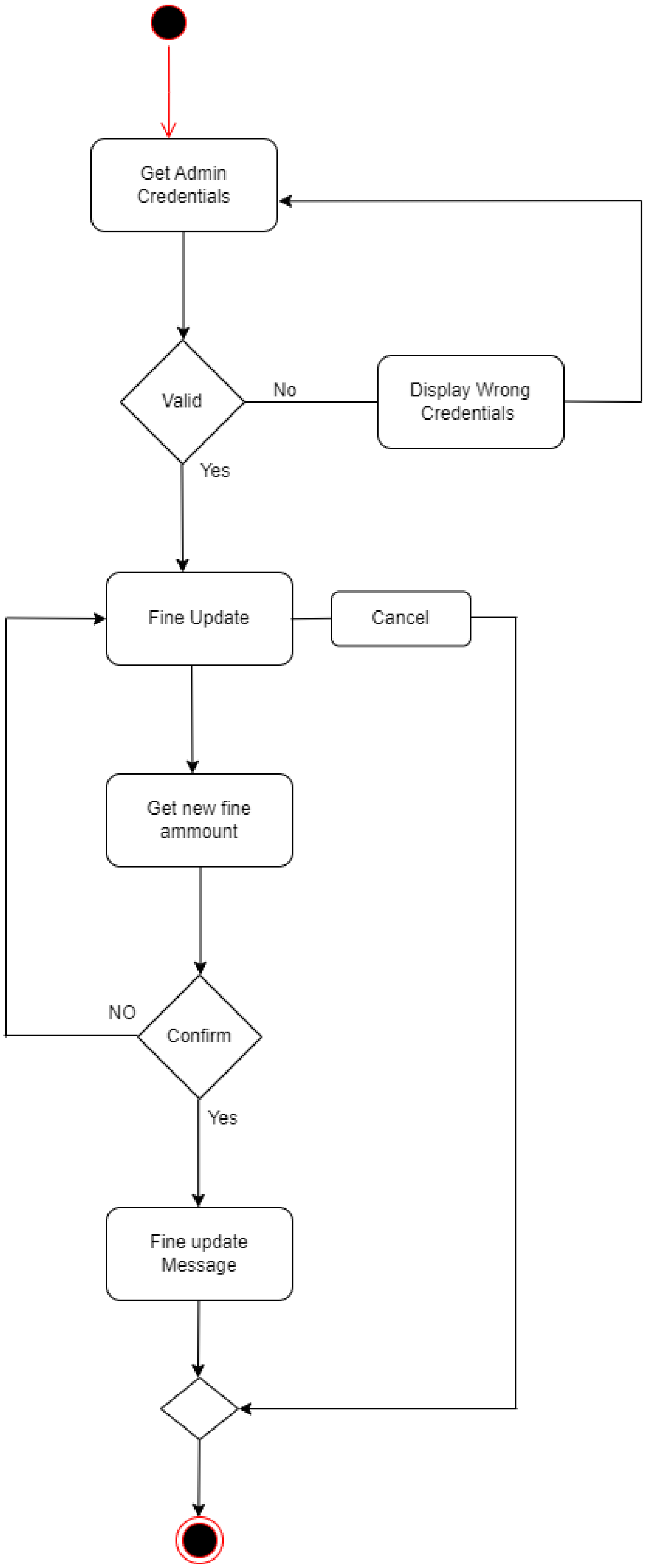


LibraSys (LMS)

Sent Notification

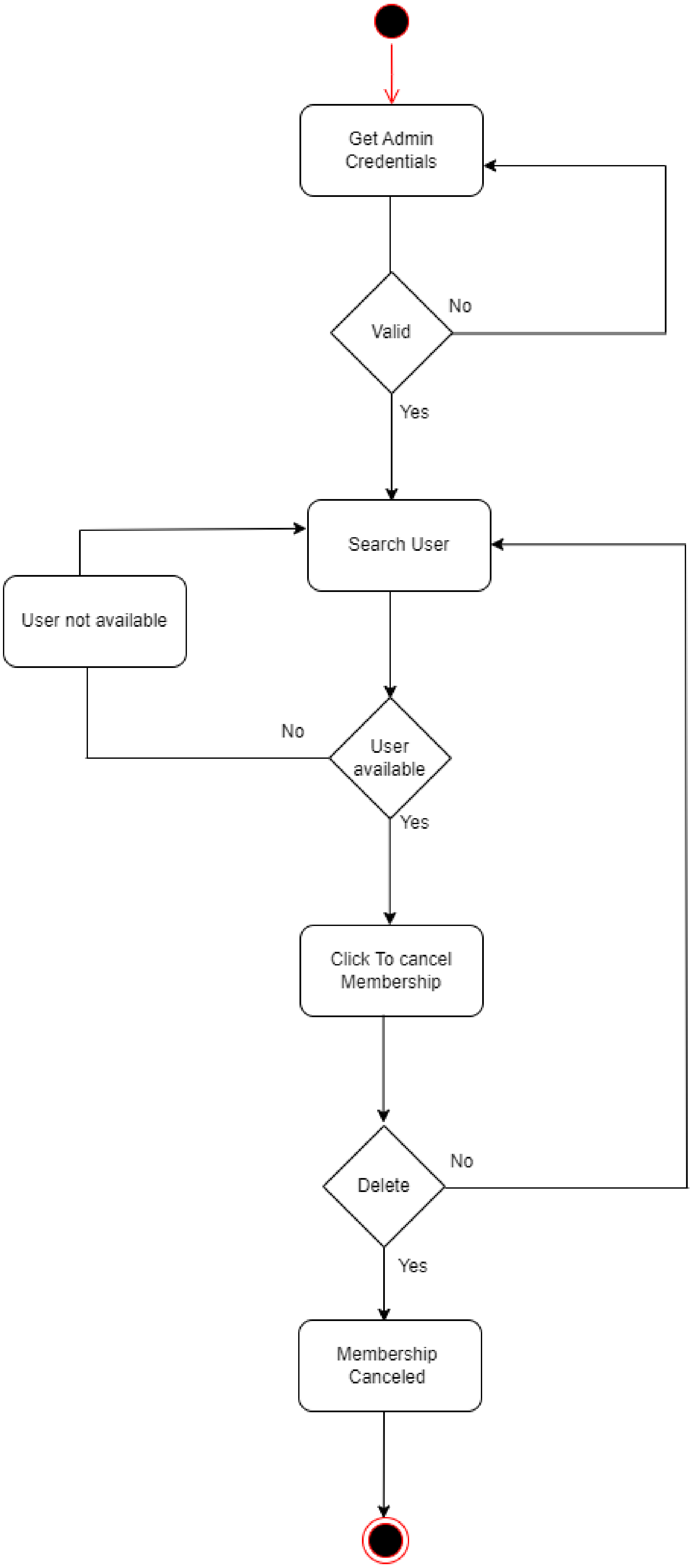


Update Details

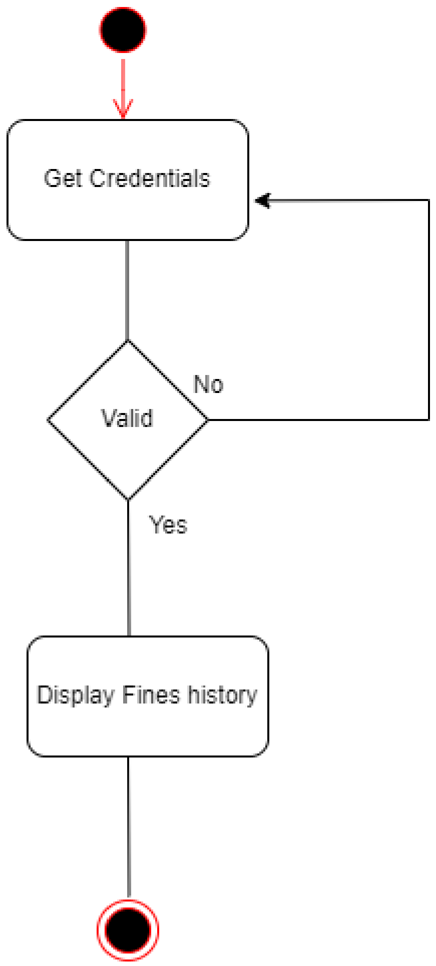


LibraSys (LMS)

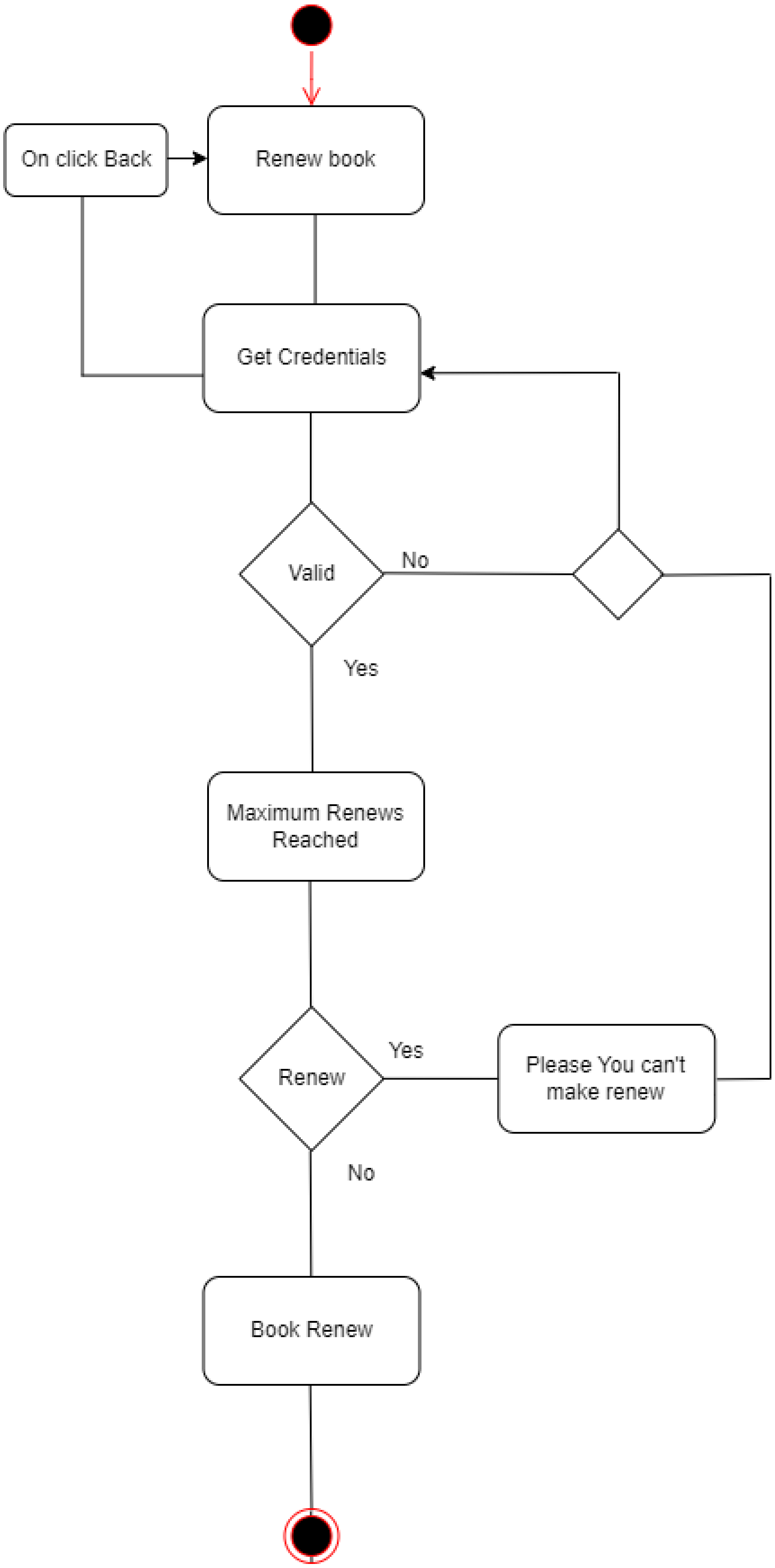
Cancel membership



Fines

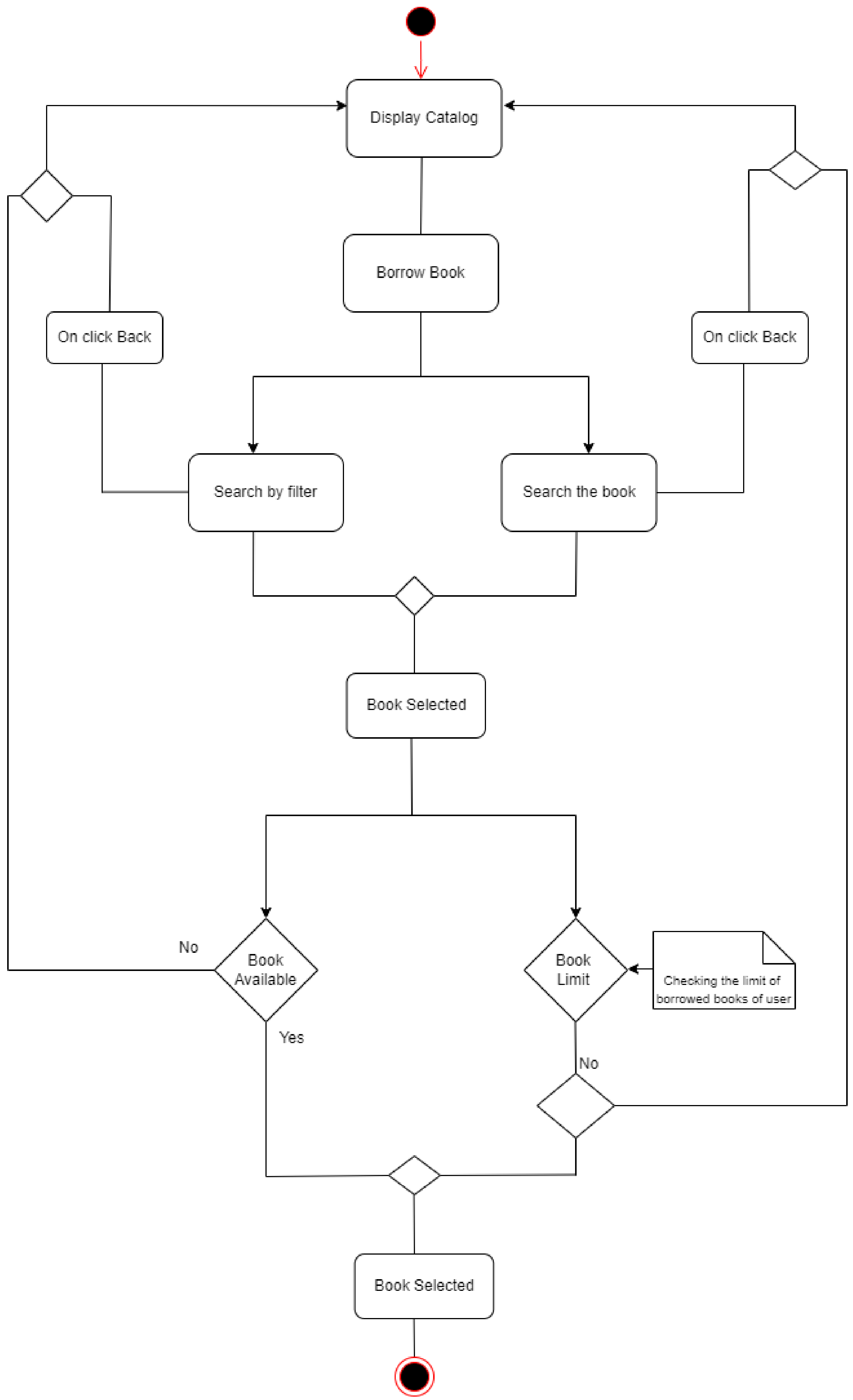


Renew book


Renew book
On click Back
Get Credentials
Valid
No
Yes
Maximum Renews Reached
Renew
Yes
Please You can't make renew
No
Book Renew

LibraSys (LMS)

Return


Display Catalog
Borrow Book
On click Back
On click Back
Search by filter
Search the book
Book Selected
No
Book Available
Yes
Book Limit
Checking the limit of borrowed books of user
No
Book Selected