AIR UNIVERSITY

SOFTWARE DESIGN SPECIFICATION

LIBARAY MANGEMENT SYSTEM

2024

PROJECT TEAM:

1) NOOR MALIK (230964)

2) GHULAM QADIR (230970)

SUBMITTED TO:

Faryal Farooq

+92 33467128

AULibrary@gmail.com

www.AULibrary.com

Document Information

Category	Information
Customer	AIR UNIVERSITY, ISLAMABAD.
Project	Library Management System
Document	Requirement Specifications
Document Version	1.0
Identifier	A36-2024
Status	Draft
Author(s)	Noor Malik and Ghulam Qadir
Approver(s)	PM
Due Date	Dec. 30, 2024
Document Location	
Distribution	1. Advisor 2. PM 3. Project Office

Software Design Specification

1. Introduction

1.1 Project Overview

The Library Management System (LMS) is a software solution designed to streamline and modernize library operations. The project focuses on providing an efficient and user-friendly platform for both librarians and library patrons. It encompasses critical functionalities such as user and book management, borrowing and returning of books, reporting and analytics, system administration, and reservation management. This document serves as the Software Requirements Specification (SRS), detailing the environment, capabilities, and requirements of the application.

The primary goals of the LMS include:

1. Enhancing operational efficiency.
2. Providing a secure and reliable system for library information.
3. Reducing operational costs while improving user satisfaction.

1.2 Existing Examples / Solutions

Numerous library management systems exist in the market, each with varying capabilities. Examples include:

- **Koha**: An open-source LMS offering comprehensive features like cataloging, user management, and reporting.
- **Aleph**: A proprietary system with robust inventory and user management functionalities.
- **Evergreen**: Known for its scalability and community-driven support.

The LMS will combine the strengths of these systems while introducing a unique feature set tailored to the client's specific needs, including integration with a LAN-based environment.

1.3 Business Scope

The LMS aims to revolutionize library operations in the following ways:

1. **Efficiency**: Streamline user registration, catalog management, and transaction tracking.
2. **Scalability**: Support growing user bases and expanding book inventories.
3. **Analytics**: Provide actionable insights through reporting and analytics tools.
4. **Cost Reduction**: Optimize resource utilization and reduce manual work.
5. **User Experience**: Improve accessibility and convenience for both staff and patrons.

1.4 Useful Tools and Technologies

The following tools and technologies will be employed:

- **Programming Languages**: Java, Python
- **Frameworks**: Spring Boot (backend), React.js (frontend)
- **Database**: MySQL for relational data management

- **Version Control**: Git for source code management
- **Testing Tools**: Selenium for automated testing and JUnit for unit testing
- **Development Environments**: Visual Studio Code, IntelliJ IDEA
- **Network Infrastructure**: LAN-based system for secure and localized access

1.5 Project Work Breakdown

The project is divided into the following phases:

1. **Requirement Analysis**: Gather and document requirements.
2. **Design**: Develop system architecture and design specifications.
3. **Development**: Implement user and system functionalities.
4. **Testing**: Conduct unit, integration, and system testing.
5. **Deployment**: Set up the LMS in the client's LAN environment.
6. **Maintenance**: Provide ongoing support and updates.

1.6 Project Feasibility

1.6.1 Technical Feasibility

The project is technically feasible given the availability of modern development tools, frameworks, and skilled personnel. The LAN environment ensures robust and secure operations.

1.6.2 Operational Feasibility

The system is designed for ease of use, ensuring a smooth transition for library staff and users with minimal training requirements.

1.6.3 Economic Feasibility

The LMS will reduce manual labor costs and enhance resource utilization, leading to significant long-term savings. Initial development costs are justified by projected efficiency gains.

1.6.4 Schedule Feasibility

The project's timeline is realistic, with sufficient resources allocated for each phase to ensure timely delivery.

1.6.5 Specification Feasibility

The system aligns with the specified functional and non-functional requirements, ensuring that all client needs are met.

1.6.6 Information Feasibility

The system's design incorporates secure data handling practices, ensuring the integrity and confidentiality of library and user information.

1.6.7 Motivational Feasibility

The project aligns with the client's goal of modernizing operations, motivating all stakeholders to support its successful implementation.

1.6.8 Legal & Ethical Feasibility

The LMS complies with legal regulations, such as data protection laws (e.g., GDPR), and adheres to ethical standards.

1.7 Risk List

Key risks include:

1. **Data Security Risks**: Potential vulnerabilities during data transmission or storage.
2. **System Downtime**: Disruptions due to server or network failures.
3. **User Resistance**: Hesitation in adopting new technology.
4. **Obsolete Technology**: Ensuring the system remains compatible with future developments.

1.8 Project Timeline

The following timeline outlines key milestones:

- **Month 1**: Requirement analysis and design specifications.
- **Month 2**: Initial development of core modules (user and book management).
- **Month 3**: Completion of advanced modules (reporting, analytics, and administration).
- **Month 4**: Testing and debugging.
- **Month 5**: Deployment and user training.
- **Month 6**: Post-deployment support and maintenance.

2. Functional and Non-Functional Requirements

2.1 Functional Requirement

2.1.1 User Management

- **2.1.1.1 User Registration**
  The system must allow users to register by providing personal details such as name, email, and library card number.

- **2.1.1.2 User Login**
  Users should be able to log in securely using a username and password.

- **2.1.1.3 Profile Management**
  Users can update their profile details, including username, avatar, and contact information.

- **2.1.1.4 Access Control**
  System administrators must assign role-based access permissions for users.

2.1.2 Book Management

- **2.1 Cataloging**
  Library staff can add, edit, and delete book records with details like ISBN, title, author, genre, and publication year.

- **2.1.2.2 Search and Retrieval**
  The system should support keyword searches, advanced filtering options, and category-based searches.

- **2.1.2.3 Inventory Management**
  The system must track book availability, including reserved, on-loan, and damaged statuses.

2.1.3 Borrowing and Returning

- **2.1.3.1 Borrowing Books**
  Users can borrow books, and the system will update the inventory accordingly.

- **2.1.3.2 Returning Books**
  Users can return borrowed books, and the inventory status will change to "available."

- **2.1.3.3 Overdue Management**
  The system will notify users of overdue books and calculate any associated fines.

2.1.4 Reporting and Analytics

- **2.1.4.1 Report Generation**
  Generate detailed reports, such as the most borrowed books, overdue books, and user activity.

- **2.1.4.2 Analytics Dashboard**
  Provide a visual representation of data such as borrowing trends and popular genres.

- **2.1.4.3 Export Data**
  Allow exporting analytics and report data in formats like CSV or PDF for external use.

2.1.5 System Administration

- **2.1.5.1 User Management**
  Administrators can add, edit, or delete user accounts and modify user permissions.

- **2.1.5.2 Configuration Management**
  Adjust borrowing limits, system settings, and notification preferences.

- **2.1.5.3 Activity Monitoring**
  Track and log user and system activity for auditing and troubleshooting purposes.

2.1.6 Reservation Management

- **2.1.6.1 Book Reservations**
  Users can reserve books currently on loan or unavailable.

- **2.1.6.2 Notification System**
  Notify users when their reserved books become available for borrowing.

2.2 Non-Functional Requirements

2.2.1 Performance

- **2.2.1.1 Concurrent User Handling**
  The system should support at least 100 concurrent users without significant performance degradation.

- **2.2.1.2 Response Time**
  Book searches and other key operations should respond within 2 seconds under normal load conditions.

2.2.2 Security

- **2.2.2.1 Data Encryption**
  All user data and communications should be encrypted using industry-standard protocols such as HTTPS and AES encryption.

- **2.2.2.2 Role-Based Access Control**
  Access to administrative and sensitive functionalities should be restricted based on user roles.

- **2.2.2.3 Data Protection**
  The system must ensure that sensitive user data is not exposed to unauthorized parties, both during storage and transmission.

2.2.3 Scalability

- **2.2.3.1 User Growth**
  The system should scale to accommodate an increased number of users without performance degradation.

- **2.2.3.2 Catalog Expansion**
  The system architecture must handle a growing number of book records efficiently.

2.2.4 Reliability

- **2.2.4.1 Uptime**
  The system must maintain an uptime of at least 99.9% to avoid disruptions.

- **2.2.4.2 Backup and Recovery**
  Regular automated backups must be conducted, and recovery mechanisms should restore data within 30 minutes of a failure.

2.2.5 Usability

- **2.2.5.1 Intuitive Design**
  The interface should be easy to navigate for users of all skill levels.

- **2.2.5.2 Accessibility**
  The system must adhere to accessibility standards such as WCAG 2.1 to accommodate users with disabilities.

2.2.6 Maintainability

- **2.2.6.1 Modular Codebase**
  The system should follow modular design principles to allow easy updates, debugging, and addition of features.

- **2.2.6.2 Documentation**
  Comprehensive documentation must be provided for both developers and administrators for seamless maintenance.

2.2.7 Compliance

- **2.2.7.1 Data Protection Laws**
  The LMS must comply with regulations such as GDPR to ensure user privacy and data security.

- **2.2.7.2 Industry Standards**
  The system should align with best practices and industry standards for library management solutions.

3. Design Considerations

3.1 Assumptions and Dependencies

Assumptions:

- Users have stable internet connections to interact with the web-based application.
- Users have modern web browser to access this website.
- Users have system that relies on third-party libraries, frameworks, and services for development, deployment, and operation.

Dependencies:

- The system depends on development frameworks.
- The system relies on a database management system (e.g., MySQL, PostgreSQL) for data storage and retrieval.
- The system depends on hosting services (e.g., AWS, Azure, Heroku) and infrastructure components (e.g., servers, databases) for deployment and operation.

3.2 Risks and Volatile Areas.

- Data Security Risks
- System Downtime and Performance Issues
- Data Loss or Corruption
- User Privacy Concerns
- Inadequate User Training and Support
- Obsolete Technology and Software Dependencies
- Inventory Management Challenges

4. System Architecture

4.1 System Level Architecture

The system consists of following major modules

- User Management Module
- Catalog Management Module
- Borrowing and Returning Module
- System Administration Module
- Reporting and Analytics Module
- Administrator Module

4.2 Sub-System / Component / Module Level Architecture

4.2.1 User Management Module

- User Management Module
- Library Card Registration
- Library Card Login
- Location Tracking
- Location Sharing
- Alert Management
- Location Viewing
- Settings Configuration

4.2.2 Catalog Management Module

- Book Cataloging
- Search and Retrieval
- Inventory Management
- Item Status Tracking

4.3 Sub-Component / Sub-Module Level Architecture

4.3.1 User Management Module

4.3.1.1 Book Cataloging

- Name
- ISBN
- Title
- Author
- Genre

- Publication Year

4.3.1.2 Search and Retrieval

- Keyword Search
- Advanced Search
- Filtering Options

4.3.1.3 Inventory Management

- Add New Items
- Update Item Details
- Remove Items
- Stock Management

4.3.1.4 Book Status Tracking

- Available On Loan
- Reserved
- Lost or Damaged

5. Design Strategies

The design of the system grows as the system is implemented, but growth is deliberate and controlled. At least some requirements are usually specified in the beginning but it is not expected to be exhaustive.

Structured Design

Structured design is mostly based on ‘divide and conquer’ strategy where a problem is broken into several small problems and each small problem is individually solved.

Top down Design

Top-down design takes the whole software system as one entity and then decomposes it to achieve more than one sub-system or component based on some characteristics. Each subsystem or component is then treated as a system and decomposed further. Top-down design

is more suitable when the software solution needs to be designed from scratch

Bottom-up Design

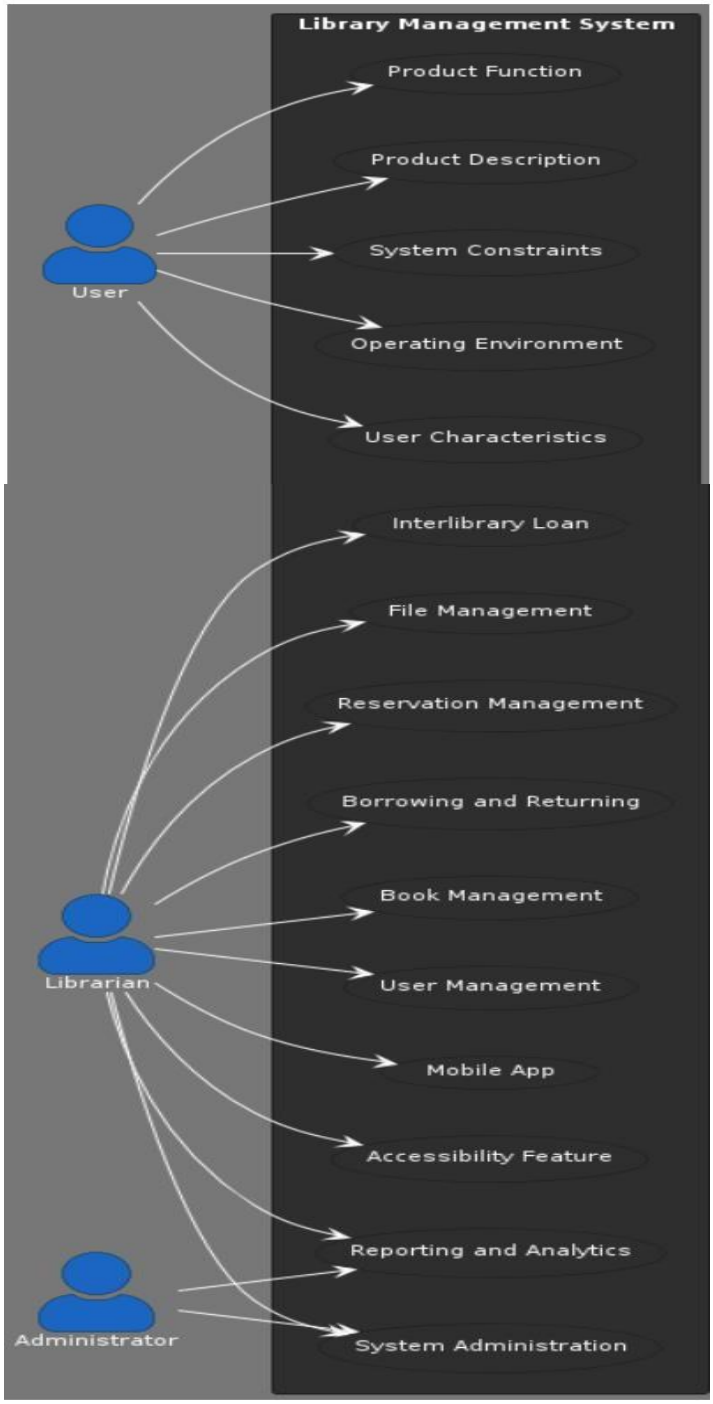
Bottom-up strategy is more suitable when a system needs to be created from some existing system, where the basic primitives can be used in the newer system.

In our project combination of above mentioned design strategies is adapted by our team.

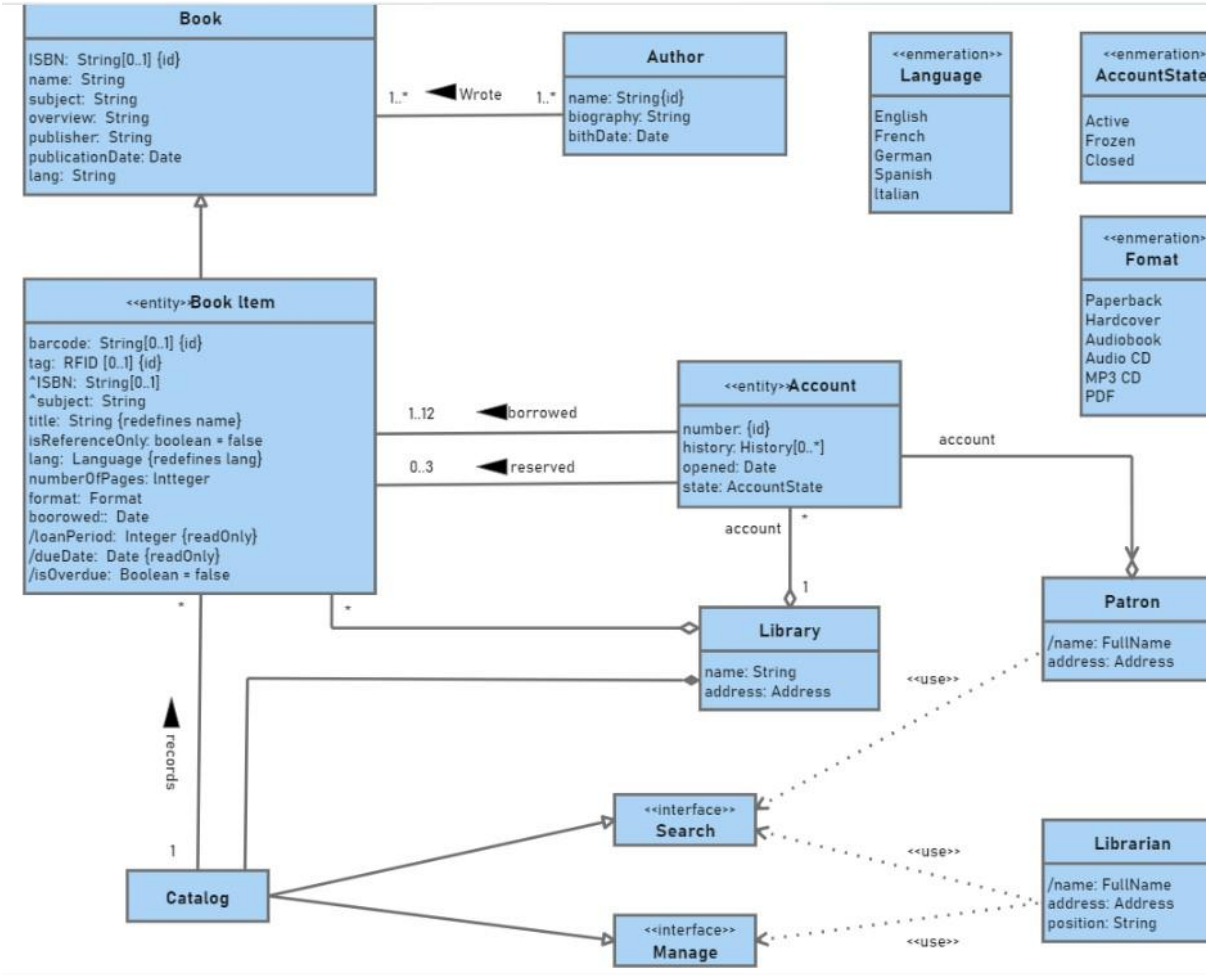
\

6. Use Detailed System Design

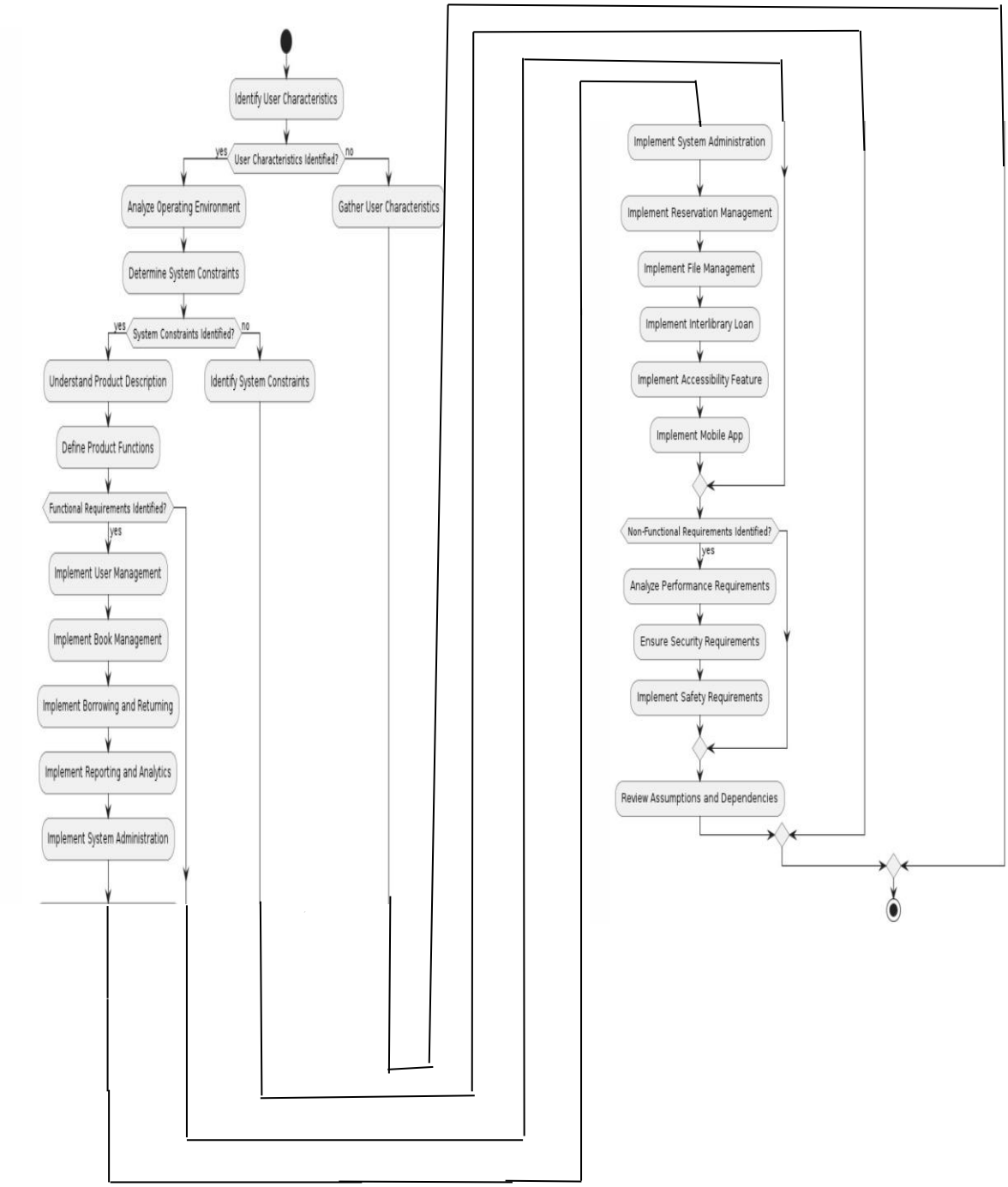
Use case diagram:



UML Diagram:

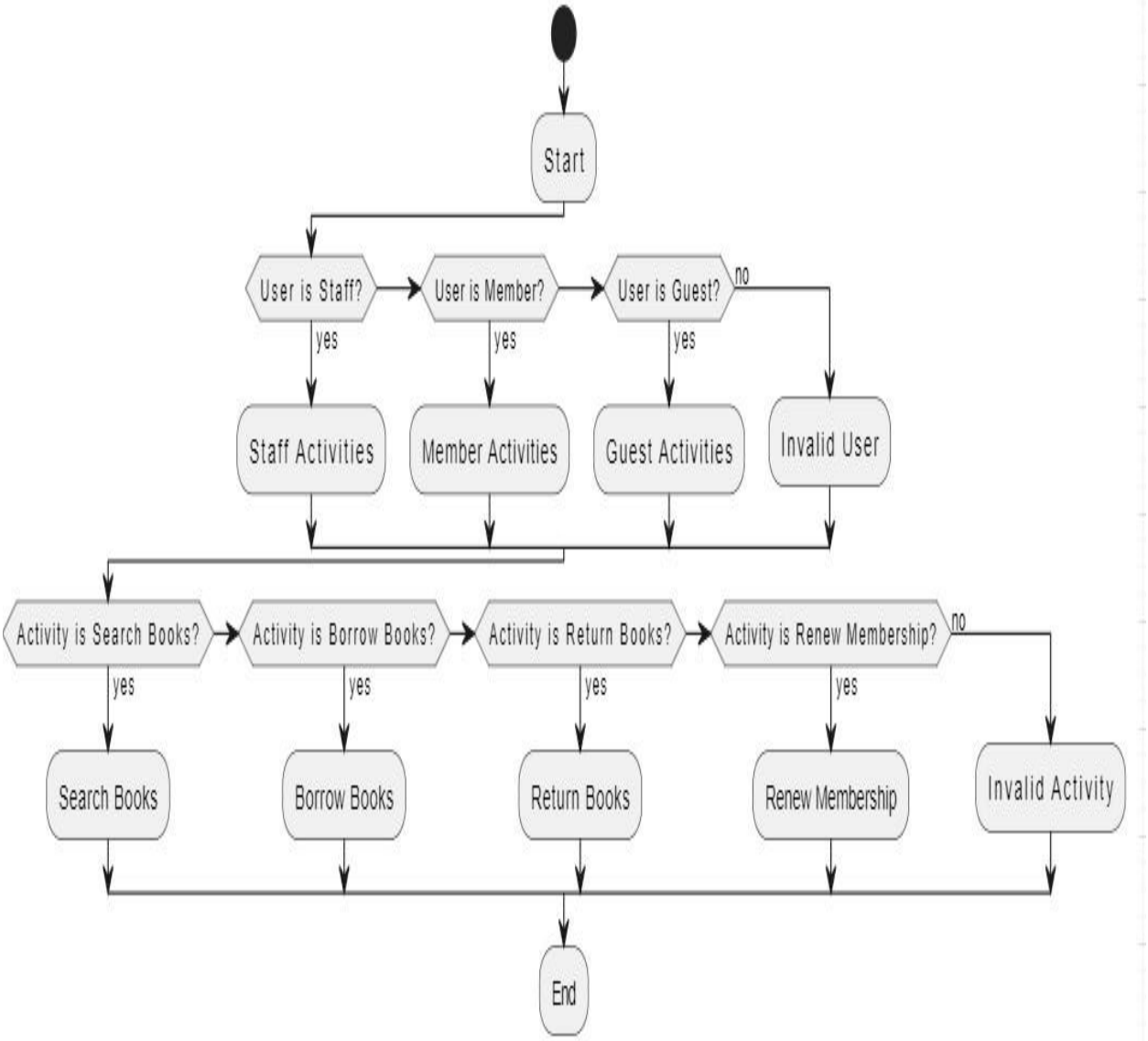

Book
ISBN: String[0..1] {id}
name: String
subject: String
overview: String
publisher: String
publicationDate: Date
lang: String
1..*
Wrote
1..*
Author
name: String{id}
biography: String
bithDate: Date
<<enmeration>>
Language
English
French
German
Spanish
Italian
<<enmeration>>
AccountState
Active
Frozen
Closed
<<enmeration>>
Fomat
Paperback
Hardcover
Audiobook
Audio CD
MP3 CD
PDF
<<entity>>Book Item
barcode: String[0..1] {id}
tag: RFID [0..1] {id}
^ISBN: String[0..1]
^subject: String
title: String {redefines name}
isReferenceOnly: boolean = false
lang: Language {redefines lang}
numberOfPages: Intteger
format: Format
boorowed:: Date
/loanPeriod: Integer {readOnly}
/dueDate: Date {readOnly}
/isOverdue: Boolean = false
1..12
borrowed
0..3
reserved
<<entity>>Account
number: {id}
history: History[0..*]
opened: Date
state: AccountState
account
account
*
1
Patron
/name: FullName
address: Address
Library
name: String
address: Address
<<use>>
records
1
<<interface>>
Search
<<use>>
Librarian
/name: FullName
address: Address
position: String
Catalog
<<interface>>
Manage
<<use>>

Activity Diagram

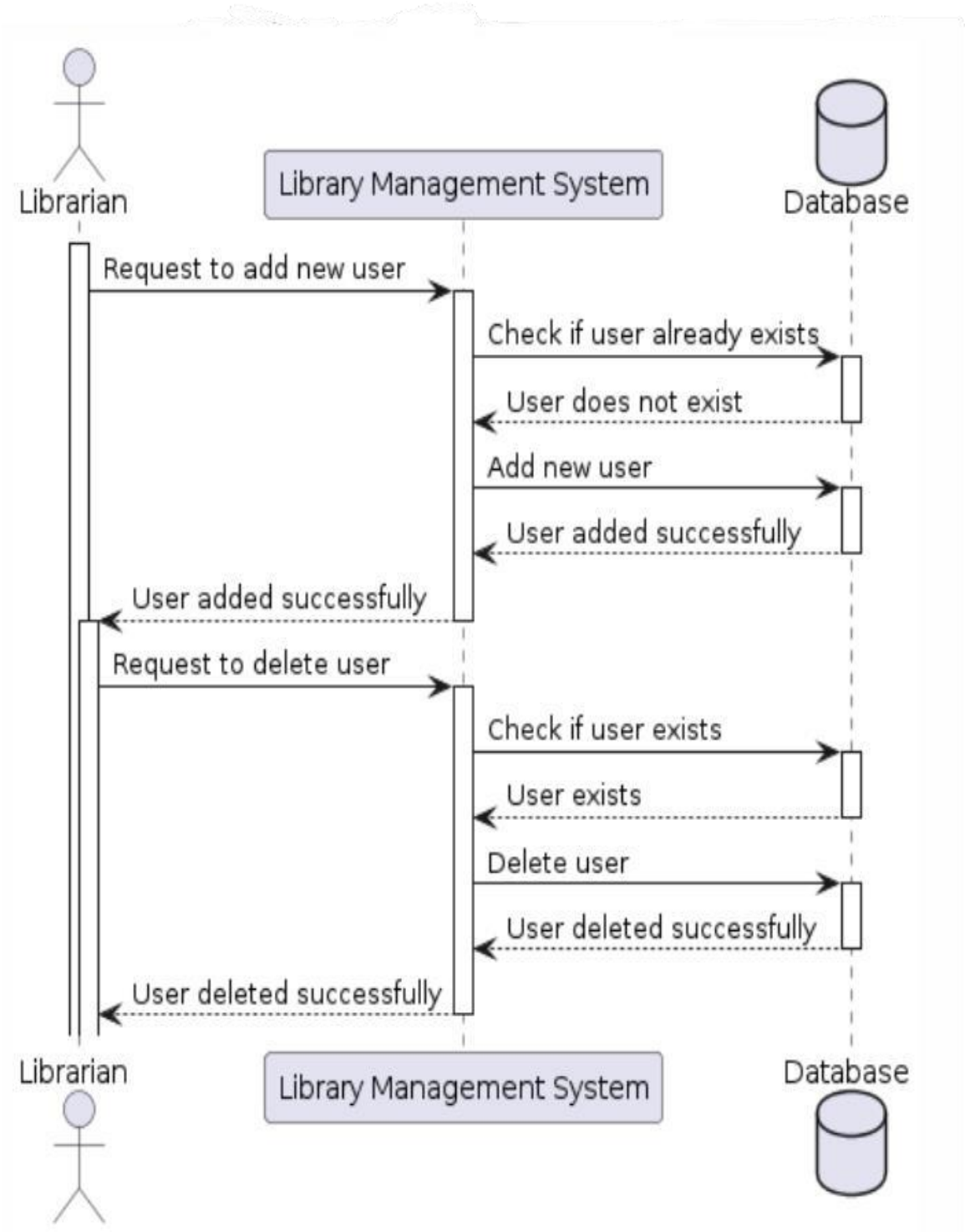


Decision-making flowchart / Process flow diagram

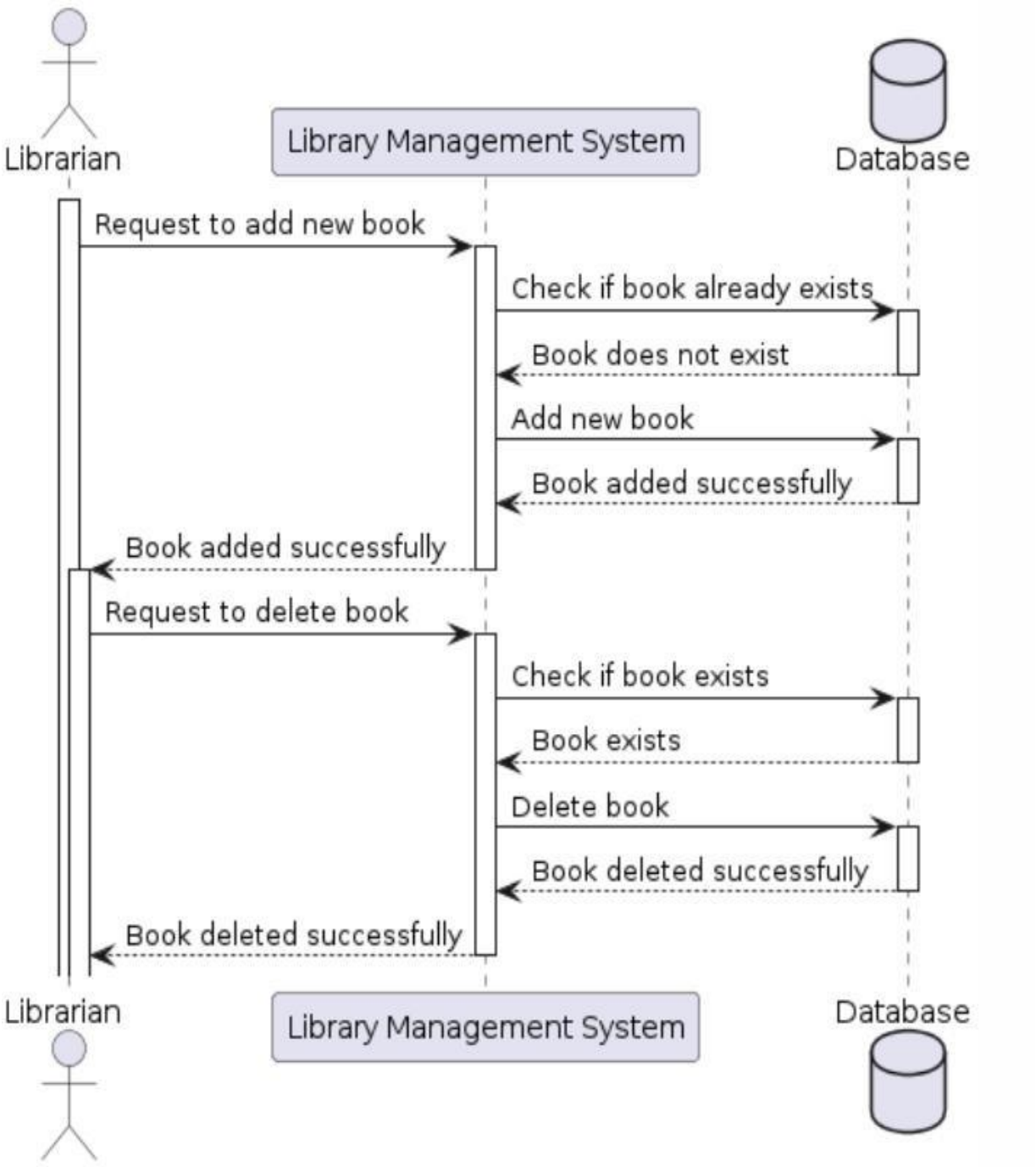
s



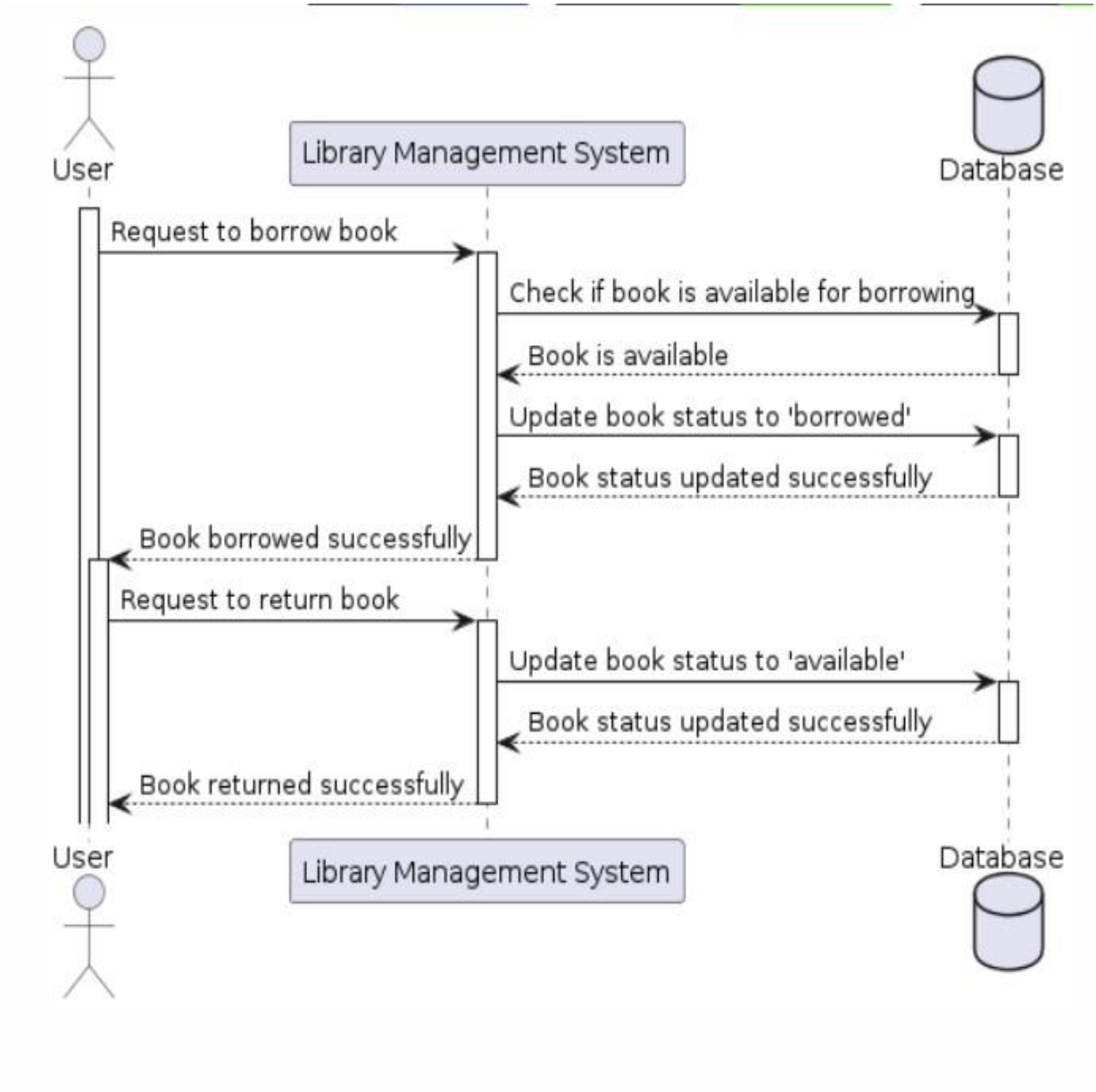
User Management Sequence Diagram



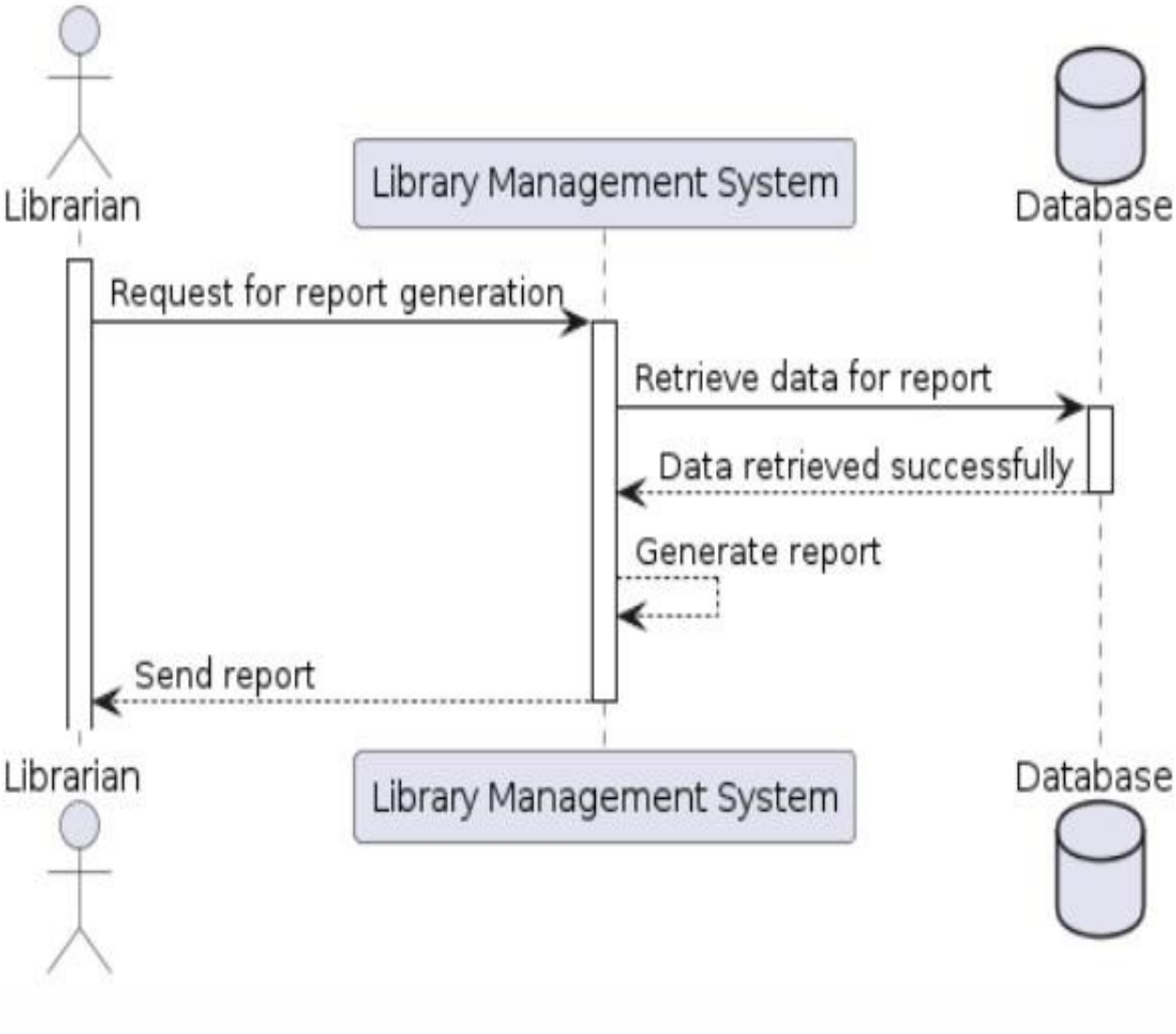
Book Management Sequence Diagram



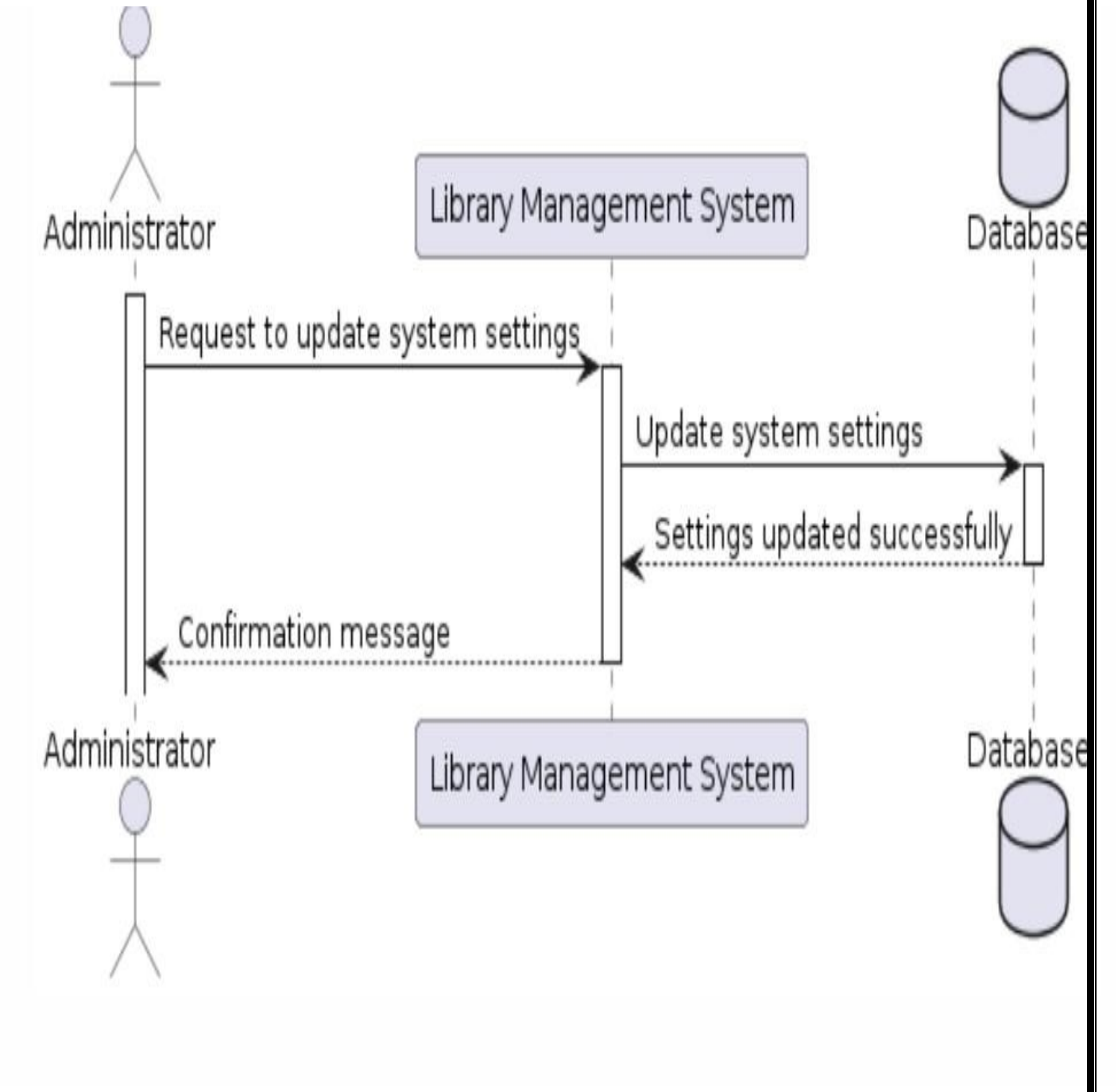
Borrowing and Returning Sequence Diagram



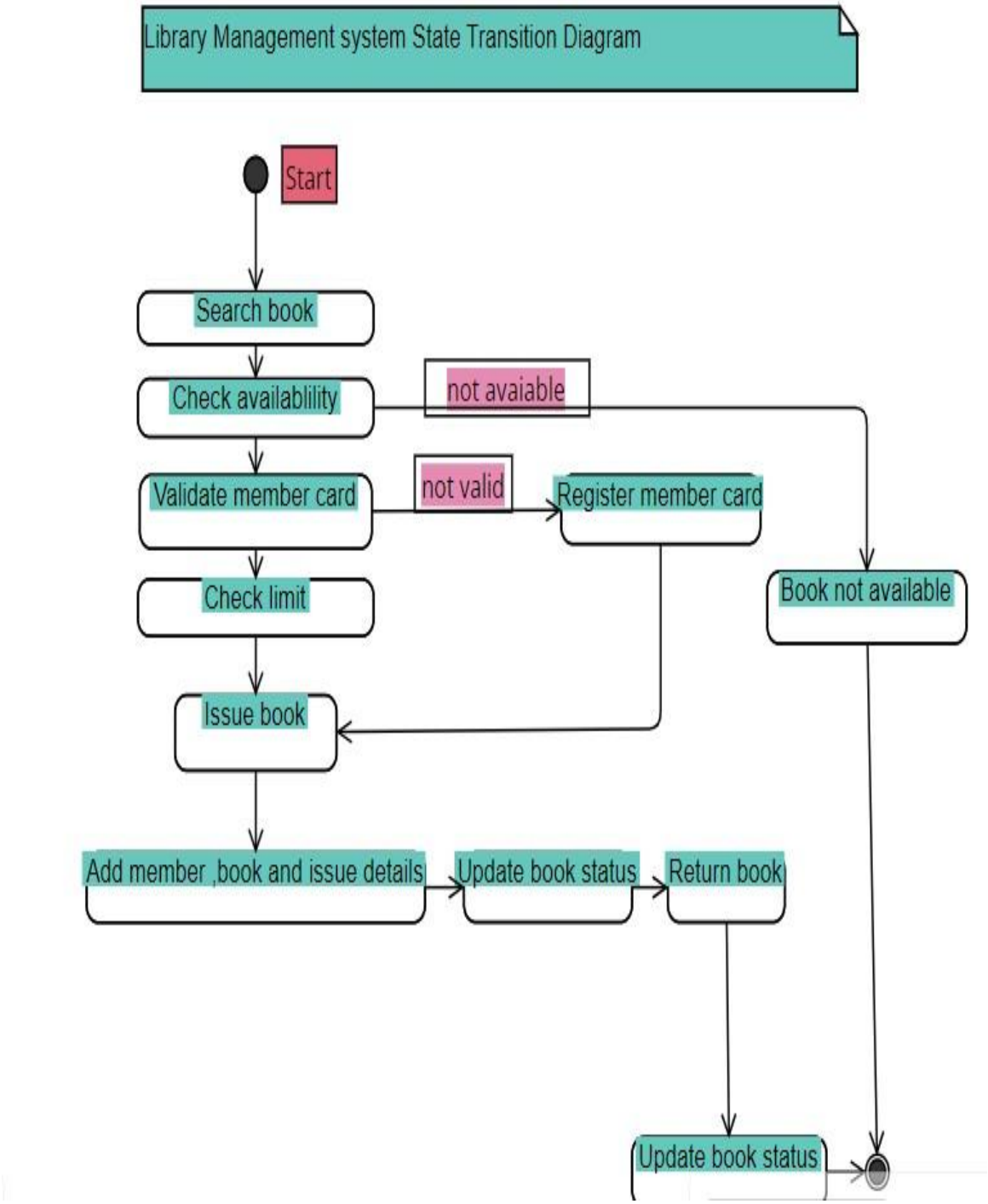
Reporting and Analytics Sequence Diagram \



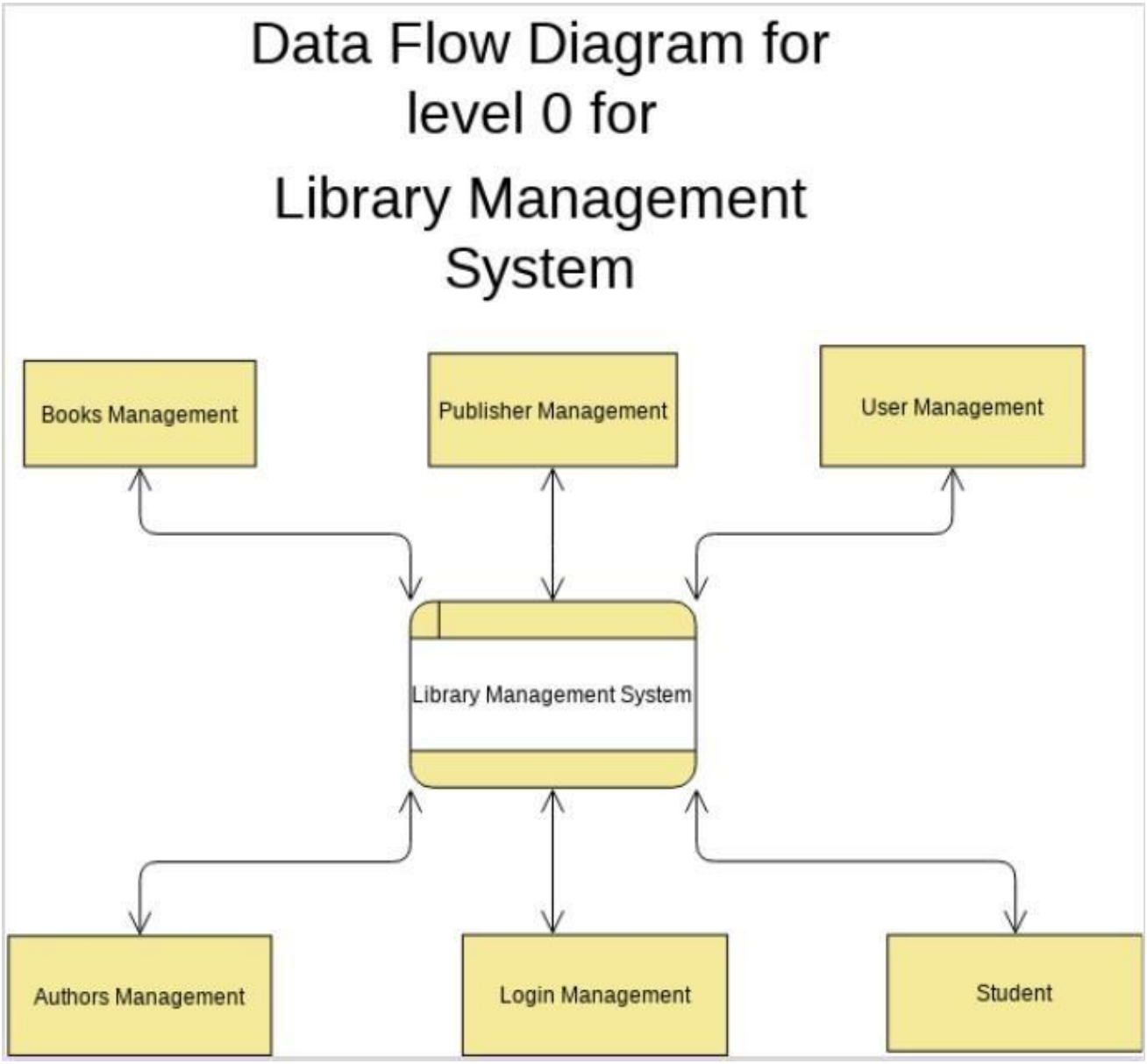
System Administration Sequence Diagram



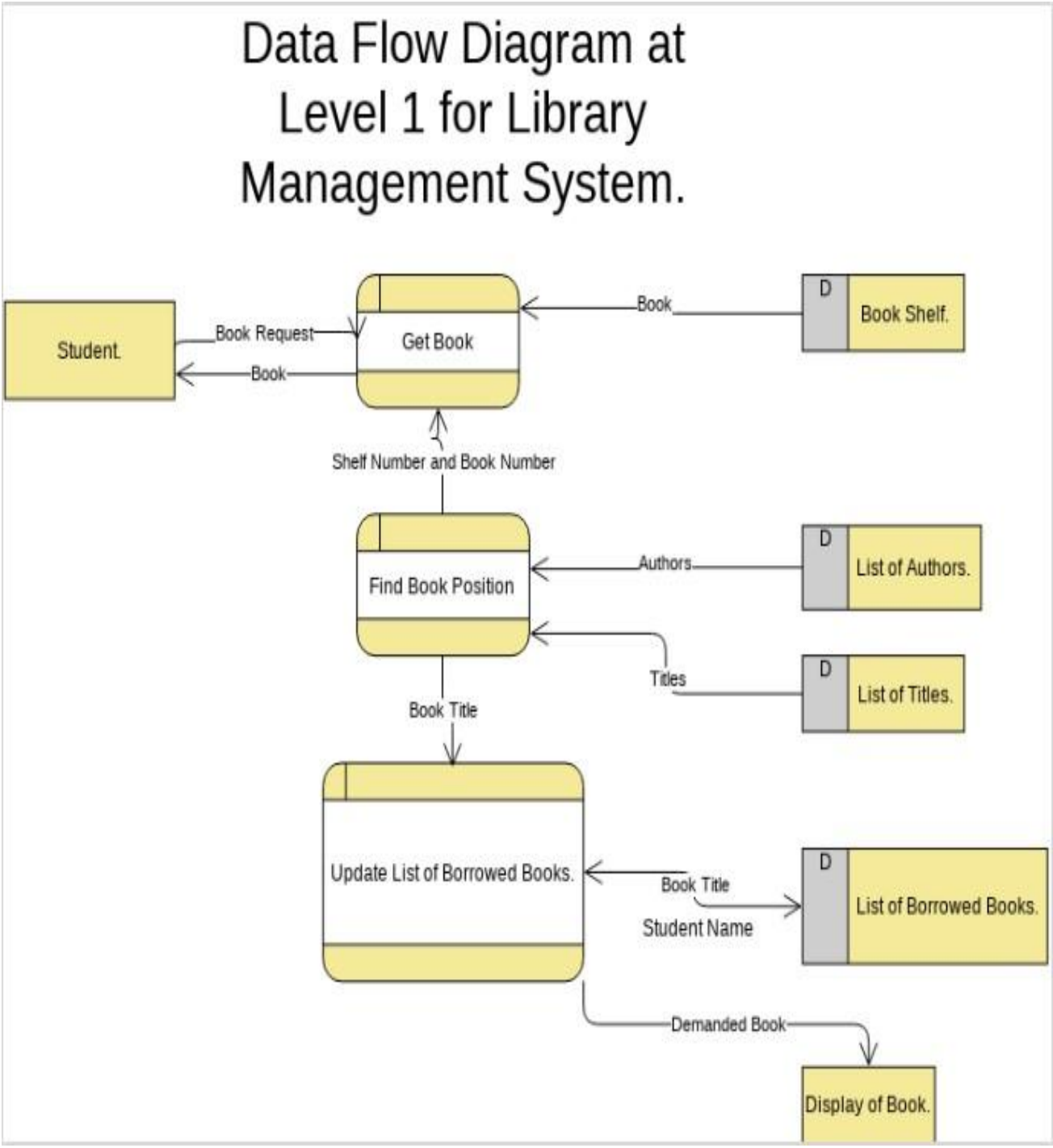
State Transaction Diagram



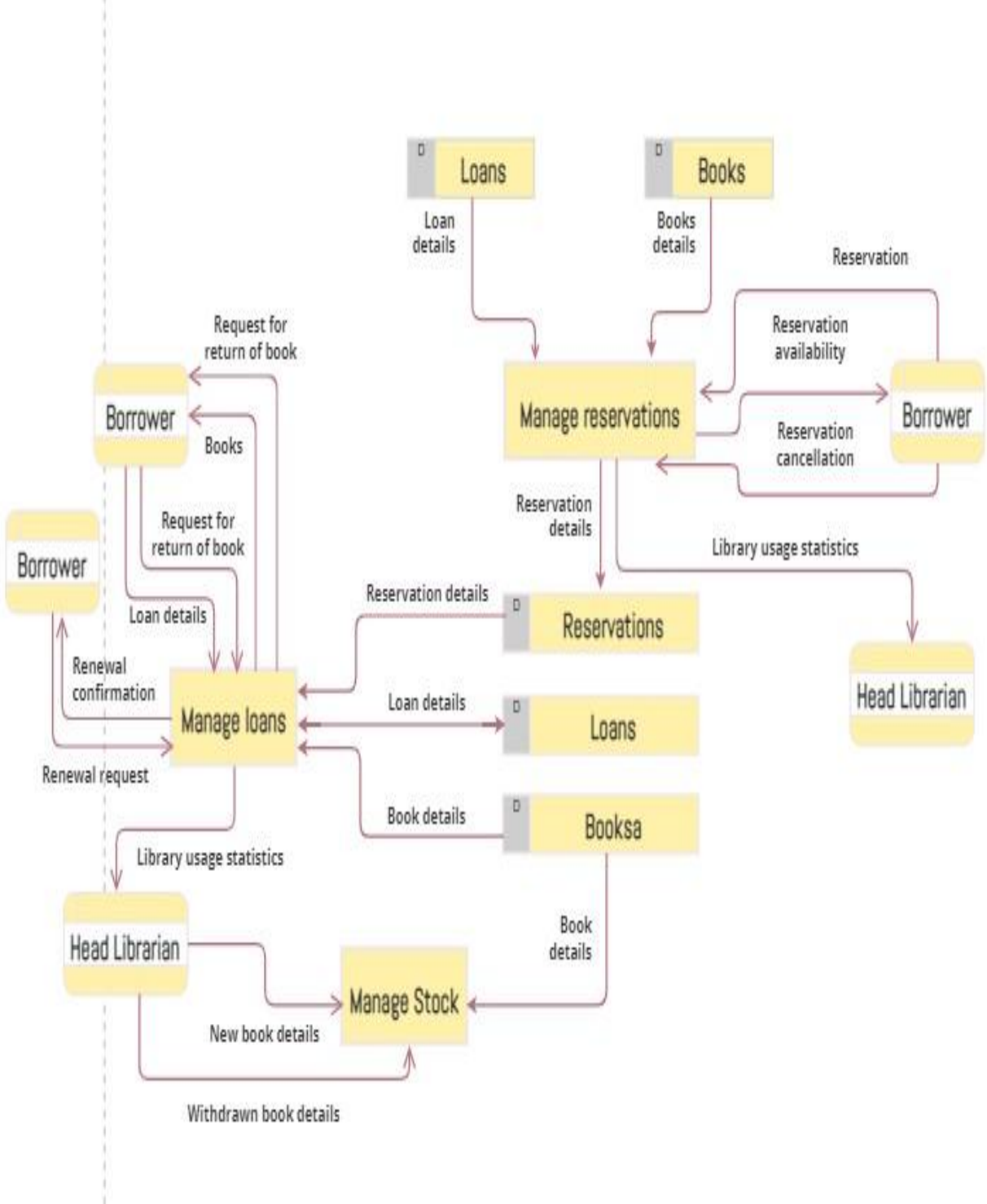
DFD Diagram Level 0



DFD Diagram Level 1

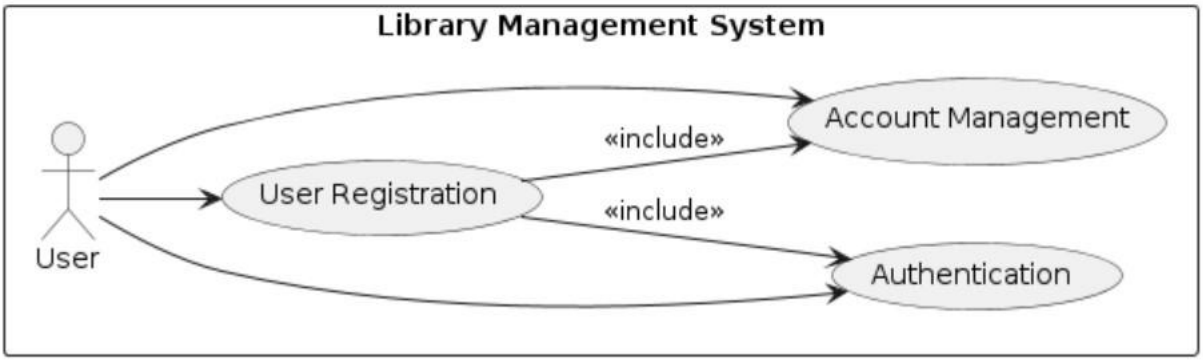


DFD Diagram Level 2 and Level 3

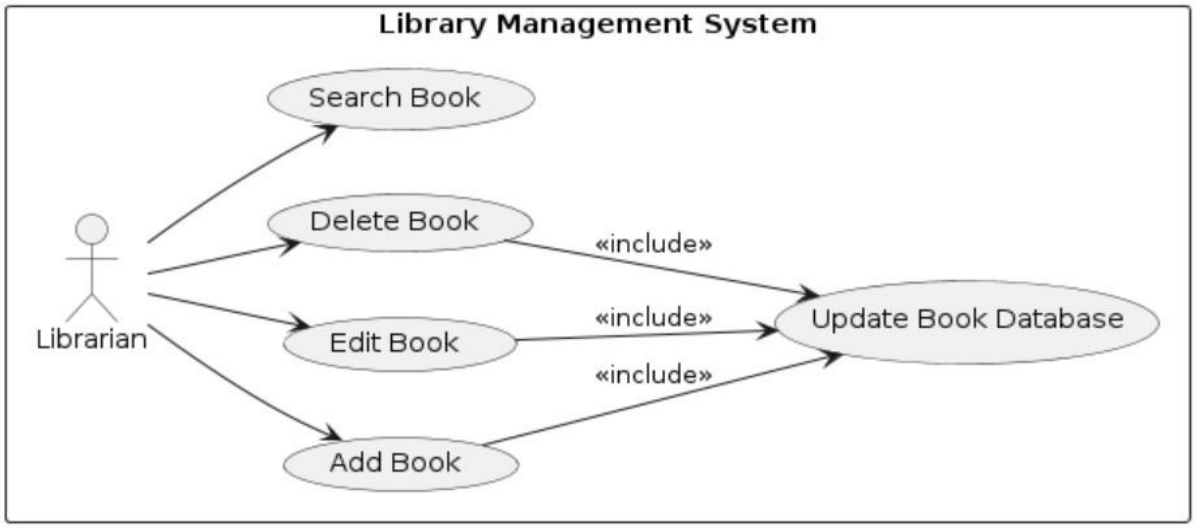


Use Cases

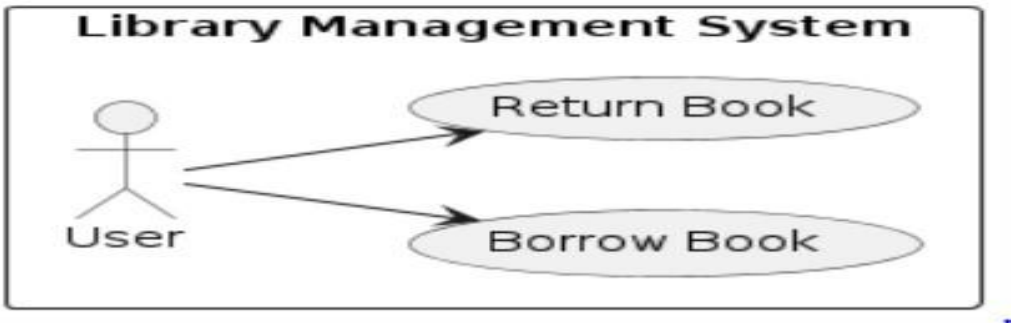
6.1 User Management



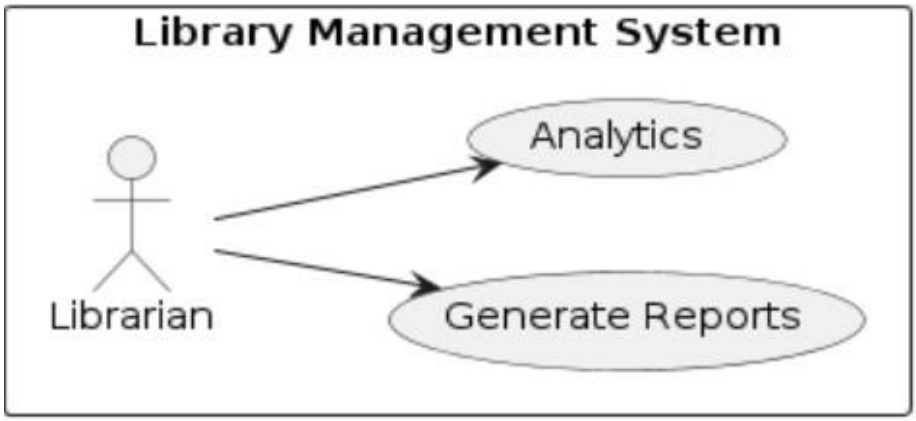
6.2 Book Management



6.3 Borrowing and Returning



6.4 Reporting and Analytics



7. Software Testing

User Management

Test Case ID:	WA-01	**QA Test Engineer:**	Ghulam Qadir
Test case Version:	1.0	**Reviewed By:**	Dr. Arif
Test Date:	30-12-2024	**Use Case Reference(s):**	UC-1
Revision History:	No Previous History		
Objective:	Validate the functionality of managing user accounts within the library system, including registration, login, profile management (such as updating username and avatar), and managing contacts.		
Product/Ver/Module:	Library Management System		
Environment:	Web-based system		
Assumptions:	1. Users have appropriate permissions and access to manage user accounts. 2. The system is operational and accessible. 3. Test data for user accounts and contacts is available for testing.		
Pre-Requisite:	1. Users have access to the library management system.		

Step No.	**Execution description**	**Procedure result**
01. 02.	User registers a new account. User enters valid details for registration.	The system prompts the user to fill in required registration The user fills in the registration form with accurate and complete information
03. 04.	User logs in with the newly registe ed account. User navigates to the pro ile management section.	The user enters their registered username and password in the login form. The user clicks on the profile or settings option within the library management system interface. The system redirects the user to the profile management page or section
Comments:		

		Passed		Failed		Not Executed

Book Management

Test Case ID:	WA-02	**QA Test Engineer:**	Noor Malik
Test case Version:	1.0	**Reviewed By:**	Dr. Arif
Test Date:	30-12-2024	**Use Case Reference(s):**	UC-2
Revision History:	No Previous History		
Objective:	Validate the functionality of managing books within the Library Management System, including adding, editing, and deleting books.		
Product/Ver/Module:	Library Management System		
Environment:	Web browser		
Assumptions:	1. Users have appropriate permissions and access. 2. The system is operational.		
	3. Test data for books is available.		
Pre-Requisite:	1. Users have access to the library management system		

Step No.	**Execution description**	**Procedure result**
01. 02. 03.	User adds a new book to the system. User edits the details of an existing book. User deletes a book from the system.	System prompts the user to fill in the details of the new book and adds it to the library. System allows the user to modify the book details and saves the changes. System removes the selected book from the library.
Comments:		

		☐	Passed	☐	Failed	☐ Not Executed

Borrowing and Returning

Test Case ID:	WA-03	**QA Test Engineer:**	Ghulam Qadir
Test case Version:	1.0	**Reviewed By:**	Dr. Arif
Test Date:	30-12-2024	**Use Case Reference(s):**	UC-3
Revision History:	No Previous History		
Objective:	Validate the functionality of borrowing and returning books within the Library Management System.		
Product/Ver/Module:	Library Management System		
Environment:	Web browser		
Assumptions:	1. Users have appropriate permissions and access. 2. The system is operational. 3. Test data for books and user accounts is available.		
Pre-Requisite:	1. Users have access to the library management system. 2. Books are available for borrowing.		

Step No.	**Execution description**	**Procedure result**
01.	User borrows a book from the library.	System allows the user to select a book and borrow it.
02.	User returns a borrowed book to the library.	System updates the book status to "returned" and removes it from the user's account.
03.	User checks the status of borrowed books in their account.	System displays the list of borrowed books in the user's account.

Comments:					
	☐	Passed	☐	Failed	☐ Not Executed

Reporting and Analytics

Test Case ID:	WA-04	**QA Test Engineer:**	Noor Malik
Test case Version:	1.0	**Reviewed By:**	Dr. Arif
Test Date:	30-12-2024	**Use Case Reference(s):**	UC-4
Revision History:	No Previous History		
Objective:	Validate the functionality of requirements management and analytics features within the Library Management System.		
Product/Ver/Module:	Library Management System		
Environment:	Web browser		
Assumptions:	1. Users have appropriate permissions and access. 2. The system is operational. 3. Test data for requirements and analytics is available.		
Pre-Requisite:	1. Users have access to the library management system. 2. Requirements and analytics features are enabled.		

Step No.	**Execution description**	**Procedure result**
01.	User creates a new requirement in the system.	System allows the user to input details and create a new requirement.
02.	User updates an existing requirement in the system.	System updates the existing requirement with the user's modifications.
03.	User views analytics data related to book borrowing trends.	System displays analytics data in graphical form.
04.	User generates a report on the most borrowed books.	System generates a report listing the most borrowed books.
05.	User analyzes the data to identify popular genres.	System presents analysis results highlighting popular genres.
06.	User exports the analytics data for further analysis.	System allows the user to download the analytics data.
07.	User deletes a requirement from th e system.	System removes the deleted requirement from the system.

Comments:						
		☐	Passed	☐	Failed	☐ Not Executed

System Administration

Test Case ID:	WA-05	**QA Test Engineer:**	Ghulam Qadir
Test case Version:	1.0	**Reviewed By:**	Dr. Arif
Test Date:	30-12-2024	**Use Case Reference(s):**	UC-5
Revision History:	No Previous History		
Objective:	Validate the functionality of system administration features within the Library Management System.		
Product/Ver/Module:	Library Management System		
Environment:	Web browser		
Assumptions:	4. Users have appropriate permissions and access. 5. The system is operational. 6. Test data for system administration tasks is available.		
Pre-Requisite:	1. Users have access to the library management system. 2. System administration features are accessible.		

Step No.	**Execution description**	**Procedure result**
01.	Admin creates a new user account i the system.	System allows the admin to input user details and create a new user account.
02.	Admin updates the permissions of a existing user account.	System updates the permissions of the existing user account as per admin's modifications.
03.	Admin monitors system usag statistics.	System displays usage statistics in graphical form.
04.	Admin configures system setting related to borrowing limits.	System allows the admin to configure borrowing limits.
05.	Admin generates a report on overdu books.	System generates a report listing overdue books.
06.	Admin exports user activity logs fo auditing purposes.	System allows the admin to download user activity logs.
07.	Admin deletes a user account fro the system.	System removes the deleted user account from the system.

Comments:

		Passed		Failed		Not Executed

8. Reference

Ref. No.	Document Title	Date of Release/ Publication	Document Source
PGBH01-2003-Proposal	Project Proposal	Sep 17, 2016	E:\Study\8th Semester\Capstone\Proposal
1	Research on Emergency Call and Location Tracking System with Enhanced Functionality for Android	Volume 3, Issue 5, May 2015	www.ijarcsms.com
2	Library Management System Using Android Application	Vol. 5 (3) , 2014, 2803-2805	www.ijcsit.com
3	Book Tracking Application– DOPE HUNT	Volume-4, Issue-ICCIN-2K14, March 2014	International Journal of Soft Computing and Engineering