Artefact Design Report

Full Name: Animesh Karmacharya

Student No: 2357899

Course: BSC(Hons) in Computer Science

University Email: A.Karmacharya@wlv.ac.uk

Supervisor: Prajwol Chhetri

Reader: Raj Pradhan

Submission: 2/02/2025

Table of Contents

Table of Figure

1. Introduction

The digital age is a working library management system that would ease operations and enhance user experience. This report details the design of an artefact for an automated library management system that was developed under the Final Year Project. The prime objective is to develop a system capable of automating some important library functions, including book borrowing, returning, recommending, and managing users.

Artefact design takes place side by side while any system is undergoing development. In other words, it refers to setting up a visual representation related to the functionalities, interactions, and workflows of systems. This report aims at visualizing the architecture, behavior, and more specifically the core functionalities of the system through different kinds of diagrams, such as Functional Decomposition, Use Case, ER, Class, and Activity Diagrams.

This report has divided the high-level, overall system-level diagrams that show the system's working at first and then moves forward with detailed artefact design for specific features. All the interactions with the system have appropriate diagrams and wireframes in each section to ensure clarity and comprehensibility. It is important to note that this report emphasizes the importance of an artefact that will provide direction on developing and implementing resilient and user-friendly artefacts for a library management system.

2. Overall System-Level Diagrams

Use Case	Authentication
Actors	User, System

a. Functional Decomposition Diagram (FDD)

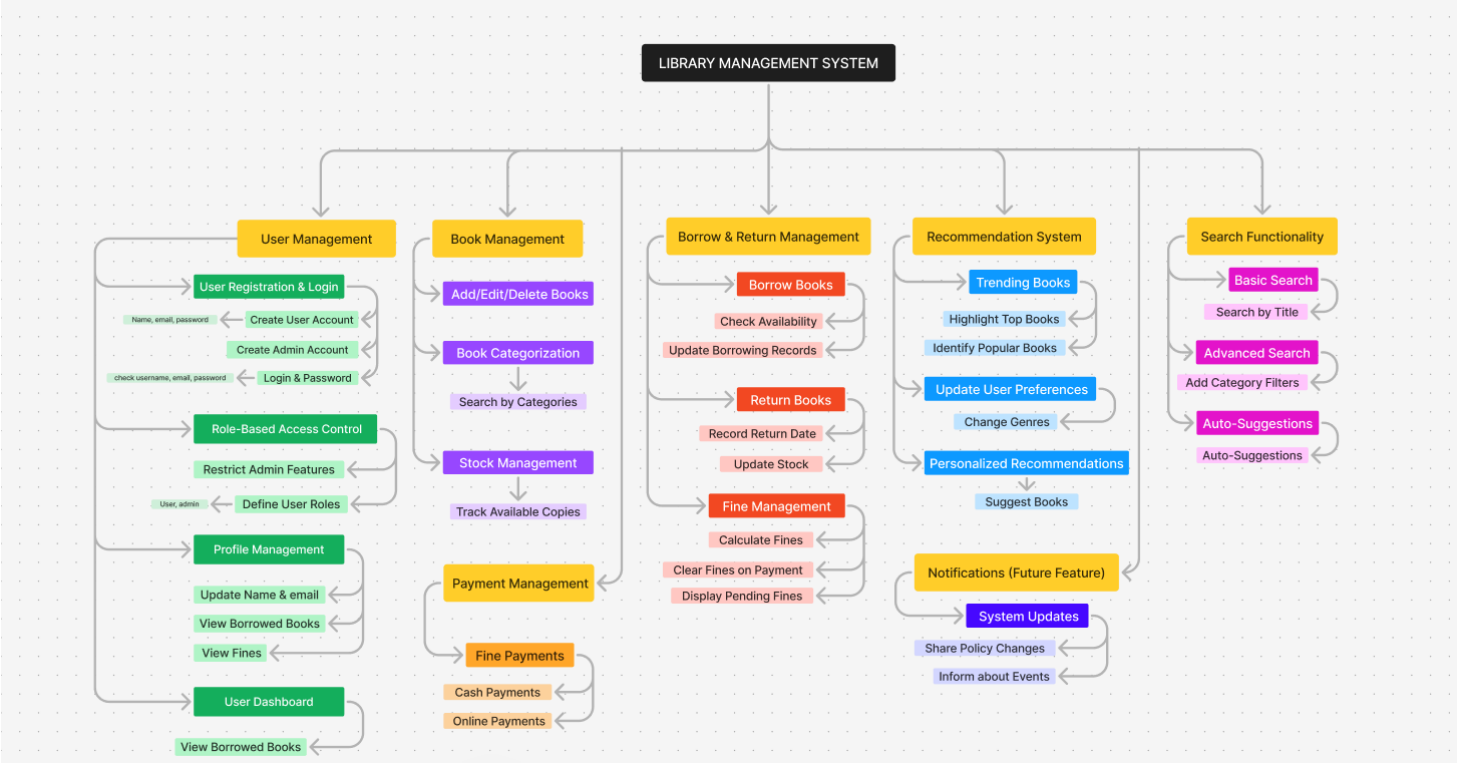

Figure 1 FDD

The FDD provides a high-level breakdown of the system's functionalities. It includes:

- **User Management:** User registration, login, and profile management.
- **Book Management:** Adding, editing, and categorizing books, along with stock management.
- **Borrow & Return Management:** Handling book borrowing, returns, and updating records.
- **Recommendation System:** Providing personalized book recommendations.
- **Search Functionality:** Enabling quick and easy book searches by title or category.
- **Payment Management:** Handling online payments for overdue fines.

b. Overall Use Case Diagram

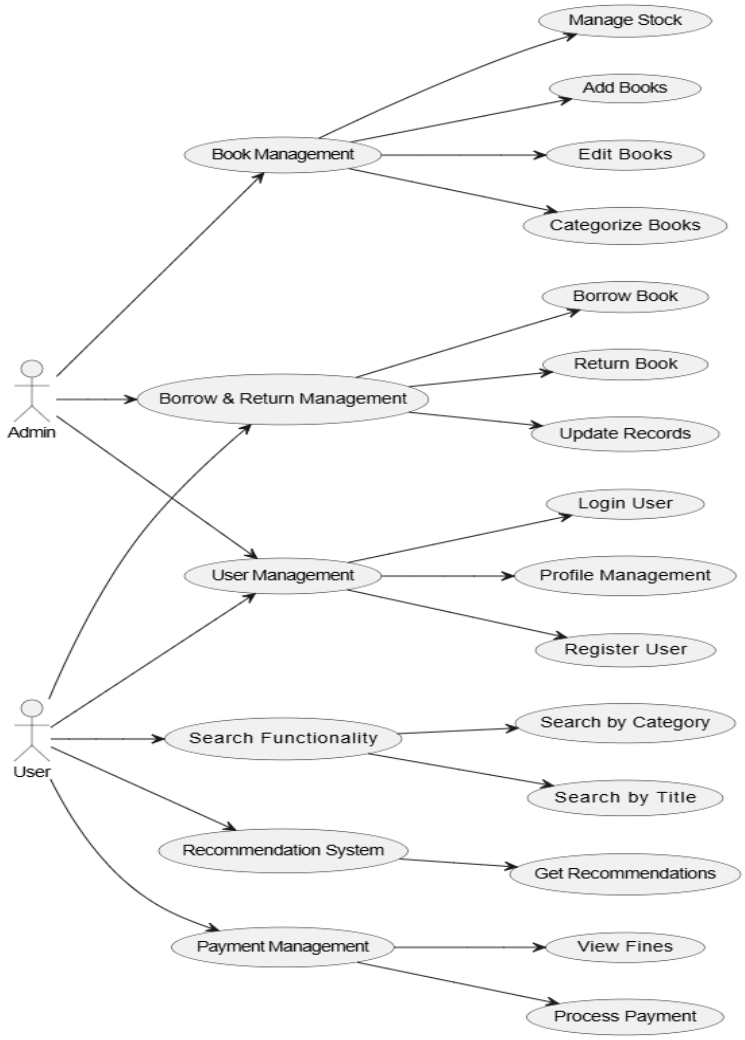

Figure 2 USE CASE

The overall use case diagram illustrates the interactions between the system's actors (users, administrators) and the system's functionalities. Key use cases include:

- User registration and login.
- Book borrowing and returning.
- Book searching and recommendations.
- Overdue fine calculations and payments.

c. Entity-Relationship (ER) Diagram

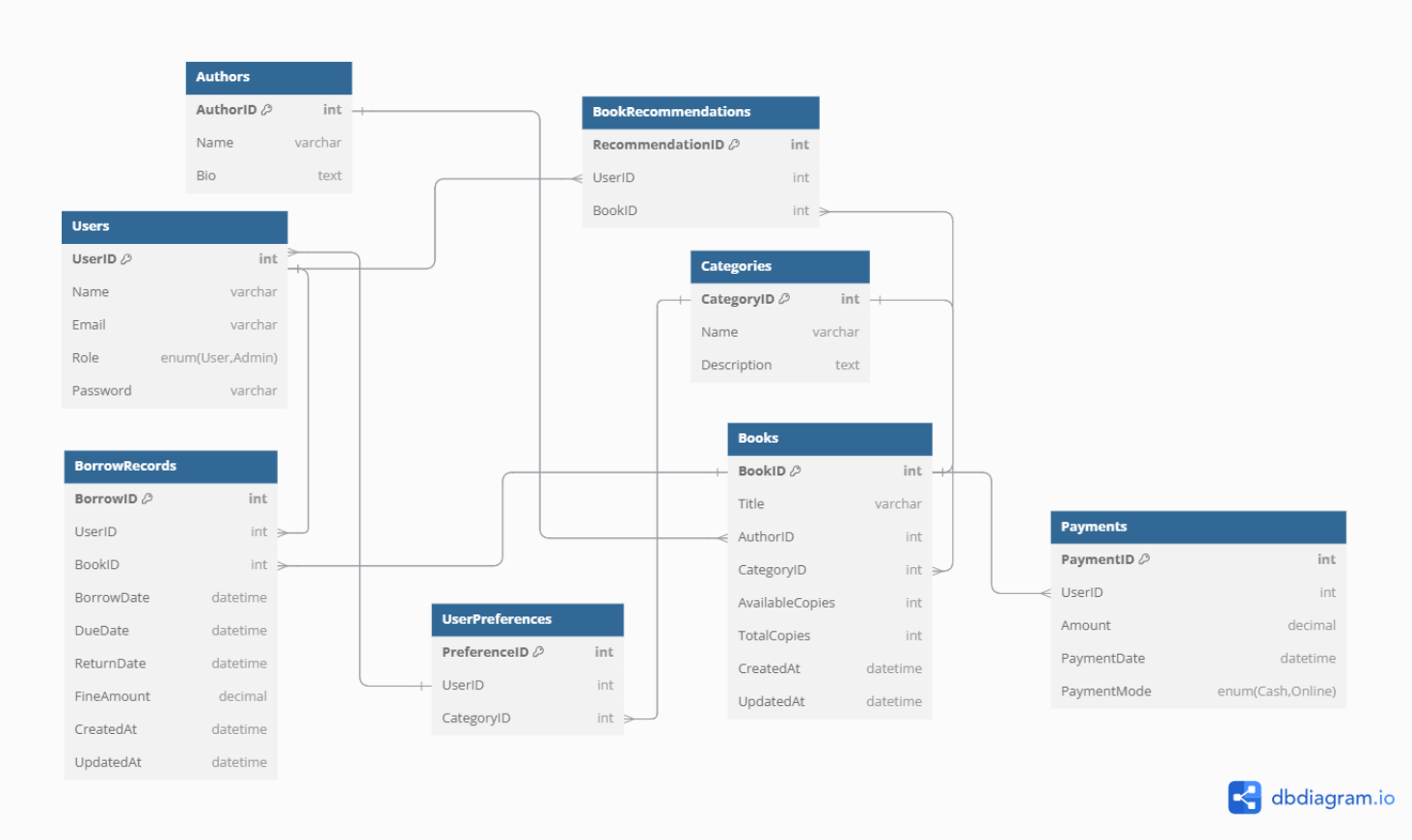

Figure 3 ER

The ER diagram represents the major entities in the system and their relationships. Key entities include:

- **User**: Stores user information such as name, email, and borrowed books.
- **Book**: Contains details like title, author, category, and availability.
- **Transaction**: Manages borrowing and returning transactions, including due dates and fines.

Key Relationships:

Users and Borrow Records: Each user can check out multiple different book items.

Users and Payments: A user has the capacity to accomplish multiple payments.

Users and Book Recommendations: A single user can obtain more than one recommendation through the system.

Users and User Preferences: Every user maintains multiple selectable preferences.

Books and Borrow Records: Books can be limited borrowed multiple times.

Books and Authors: A single author develops multiple publications in their writing career.

Books and Categories: Within a category multiple publications can be included.

Books and Book Recommendations: The recommendations system lets one book obtain several matching entries.

d. Class Diagram

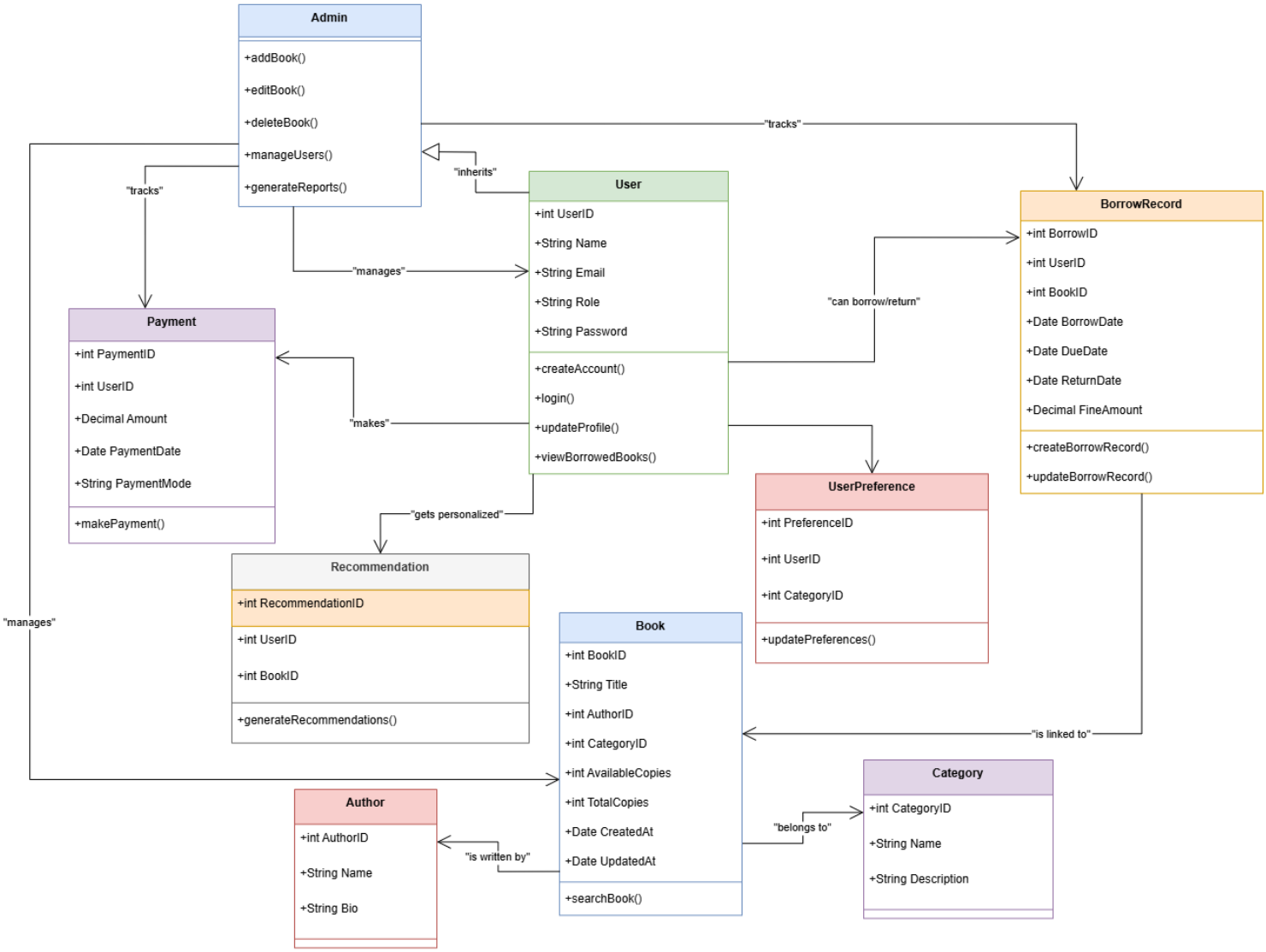

Figure 4 CLASS

The class diagram outlines the system's classes, their attributes, methods, and relationships. Key classes include:

- **User**: Manages user-related operations like registration and login.
- **Book**: Handles book-related operations like adding, editing, and searching.
- **Transaction**: Manages borrowing, returning, and fine calculations.

Key Relationships

- **Inheritance:** Admin is a specialized User with extra capabilities.
- **Associations:**
    - **User to Borrow Record:** One user can have multiple borrow records.
    - **User to Payment:** One user can make multiple payments.
    - **User to Recommendation:** One user can receive multiple recommendations.
    - **User to UserPreference:** One user can have multiple preferences.
    - **Book to Author:** One book is written by one author; an author can write many books.
    - **Book to Category:** One book belongs to one category.
    - **Borrow Record to Book:** A book can be borrowed multiple times.
    - **Category to UserPreference:** One category can be preferred by multiple users.

e. Activity Diagram

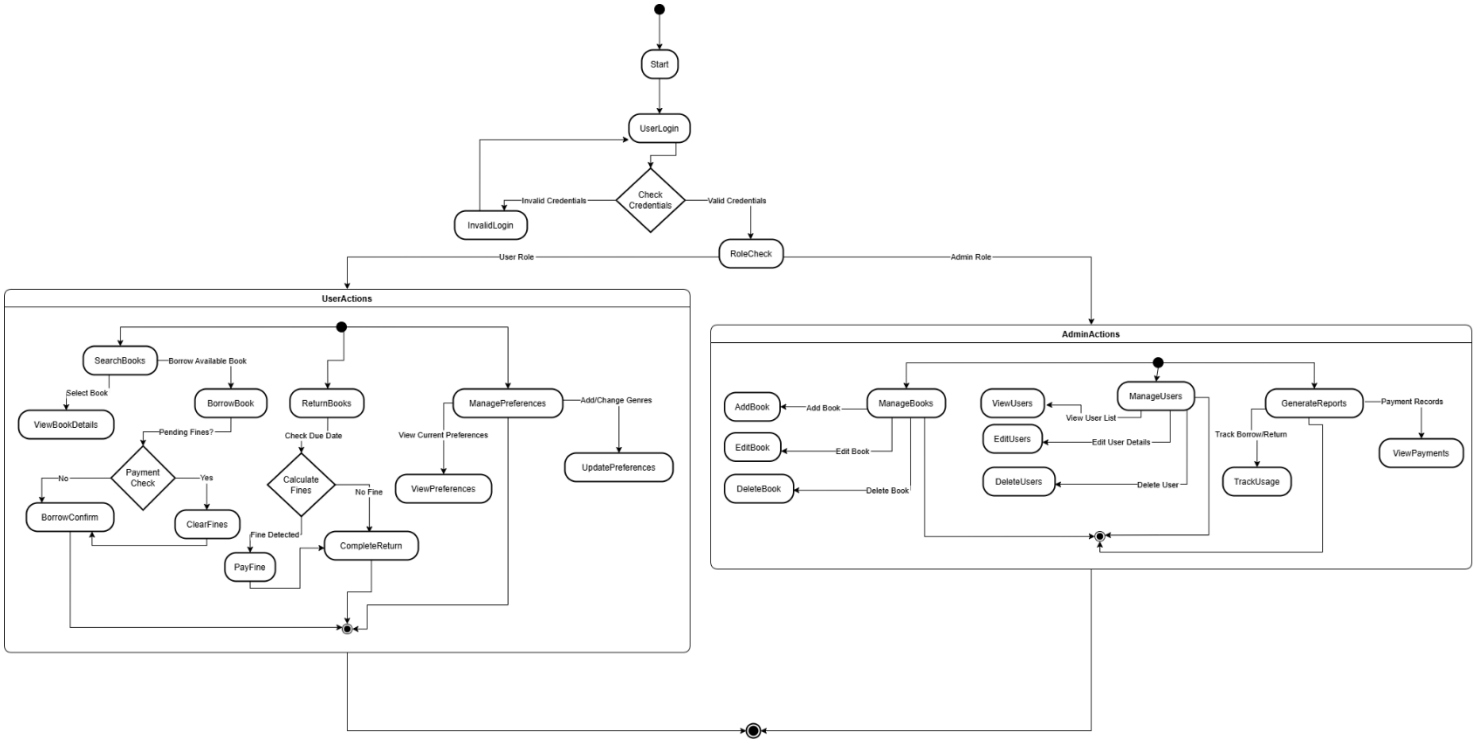

Figure 5 Activity

3. Feature -Specific Artefact Design

I. Automate Library Operations

Use Case	Automate Library Operations
Actors	User, Admin, System
Description	The system automates book borrowing, returning, and overdue fine calculations. Users can borrow books, and the system updates the records. If a book is overdue, the system calculates the fine.

a. Detailed Use Case Diagram

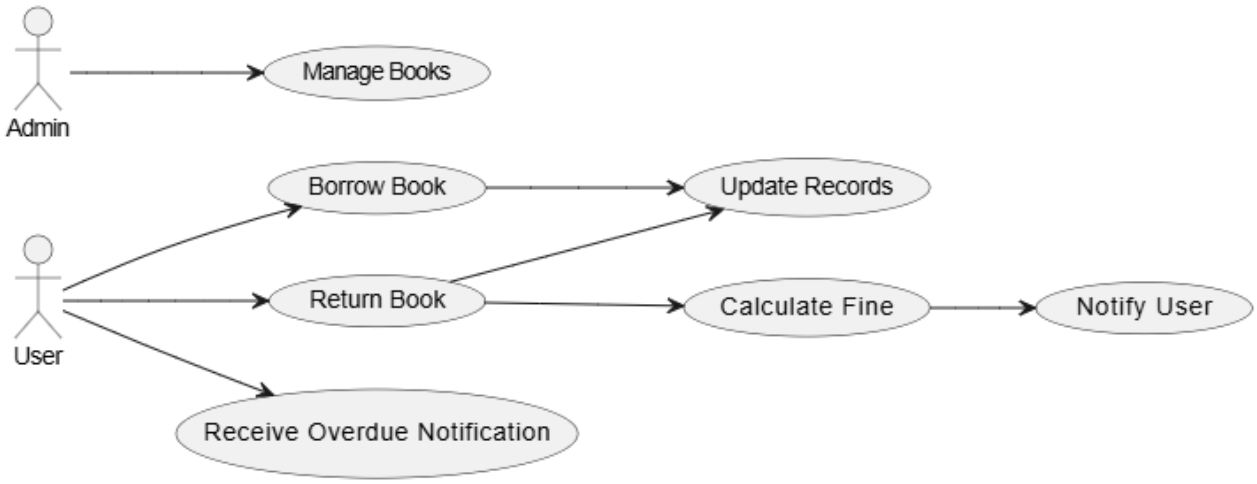

Figure 6 use case

The User can borrow, return books, and receive overdue notifications. The system updates record and calculates fines if a book is returned late. The admin manages the library's book inventory by adding, removing, or updating books. This diagram shows how Users interact with the system for borrowing and returning, while Admins handle the library's collection

b. Sequence Diagram

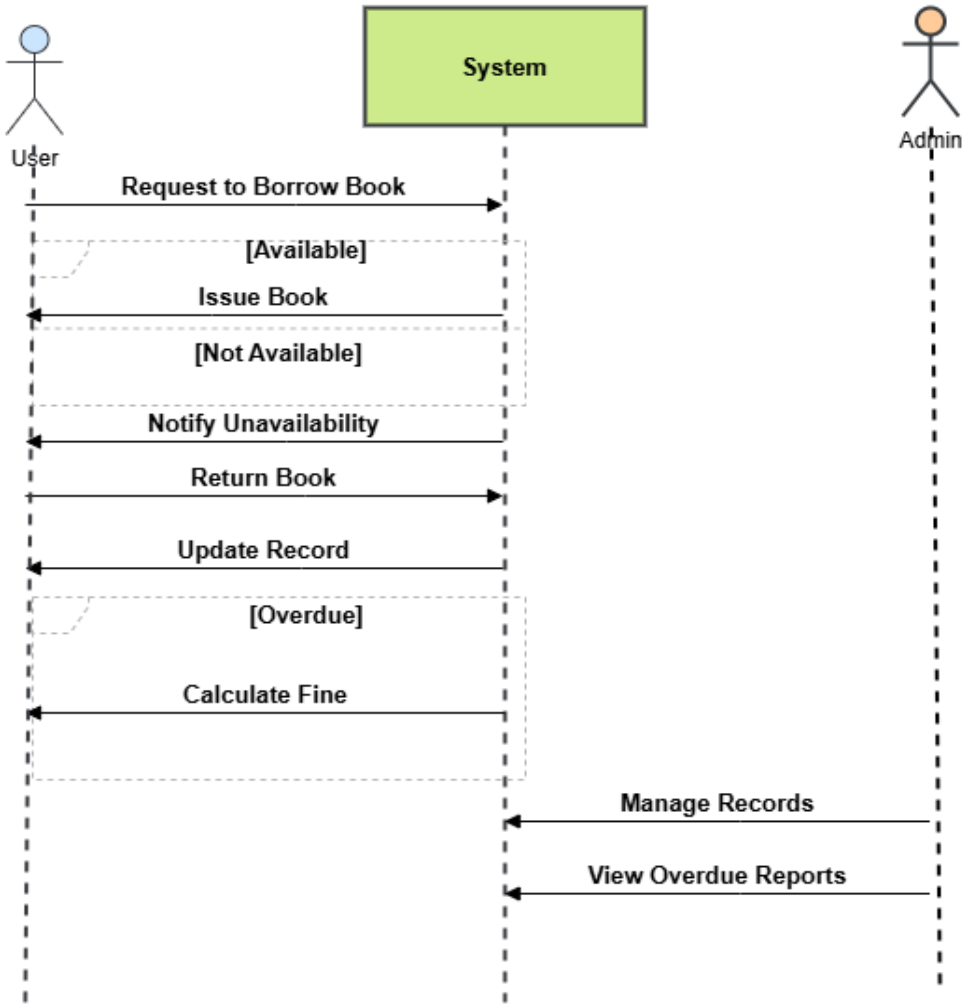

Figure 7 sequence

The key process in this feature is borrowing and returning a book. The user requests to borrow a book, and the system checks its availability. If available, the system issues the book and updates the records. When the user returns the book, the system checks if it's overdue. If overdue, the system calculates a fine and notifies the user. The admin manages records and monitors overdue reports to ensure proper management of the system.

c. Activity Diagram

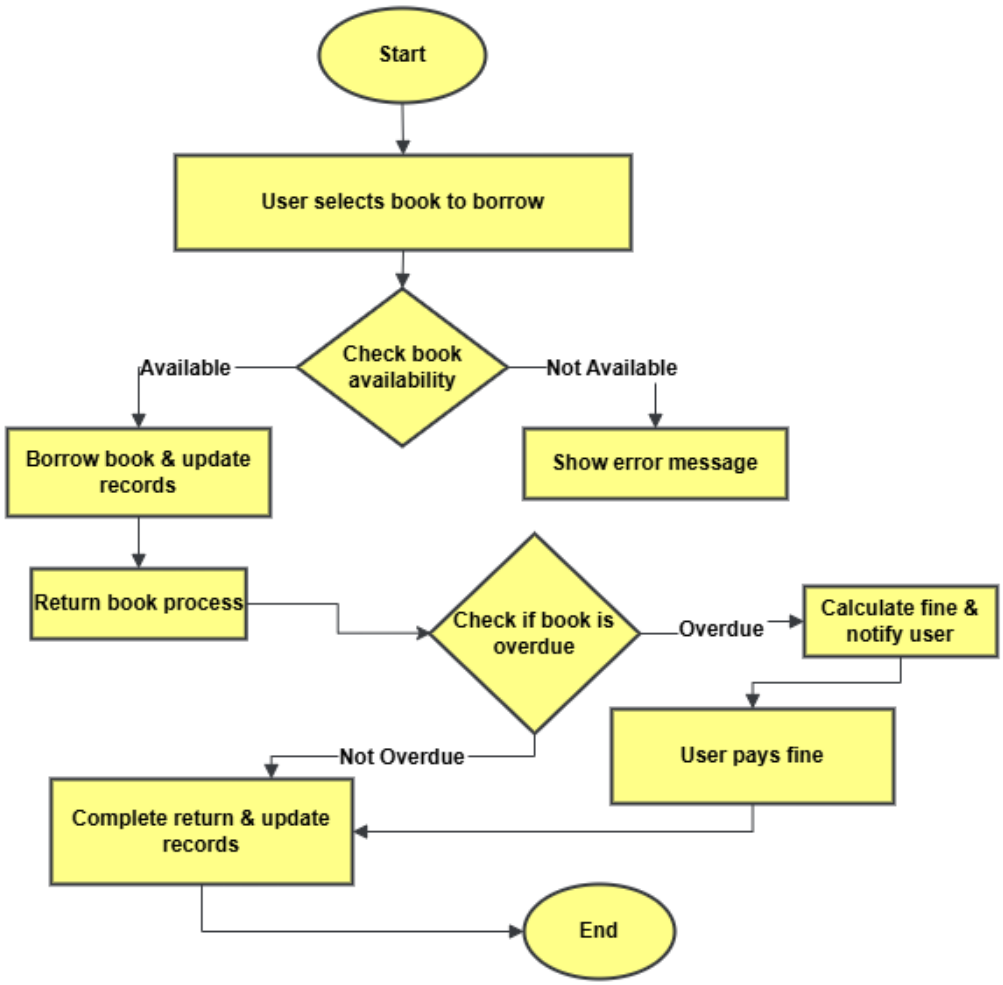

Figure 8 AD

This activity diagram outlines the book borrowing and returning process. When a user selects a book, the system checks its availability. If available, the user borrows it, and records are updated. If not, an error is shown. Upon return, the system checks if the book is overdue. If on time, the return is completed. If overdue, a fine is calculated, and the user must pay before completing the return. Once the fine is paid, the return is finalized, and the process ends.

d. Wireframe

Low fidelity design of the user interface for this feature.

II. Personalized Dashboards

Use Case	Personalized Dashboards
Actors	User, Admin
Description	The system provides personalized dashboards for users and administrators. Users can view borrowed books, due dates, and recommendations. Admins can manage books, users, and transactions.

a. Detailed Use Case Diagram

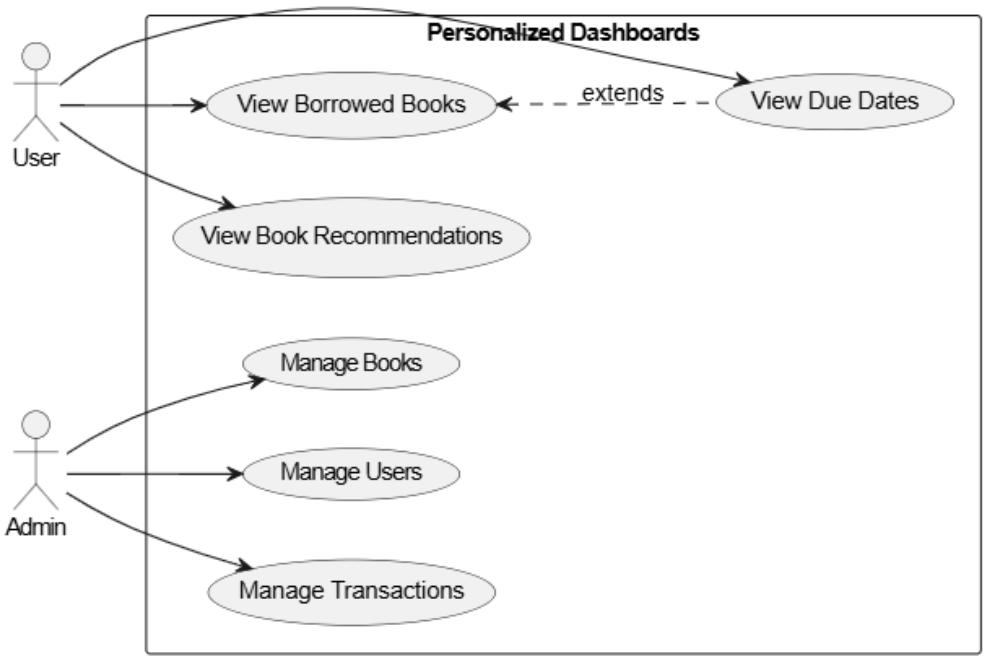

Figure 9 UC

This diagram shows two actors, User and Admin, within the "Personalized Dashboards" system. The User can view borrowed books, due dates (extended from borrowed books), and book recommendations. The Admin

can manage books, users, and transactions. The use cases are grouped for each actor, with an extension relationship between "View Borrowed Books" and "View Due Dates." The diagram clearly illustrates the system's personalized dashboard features.

b. Sequence Diagram

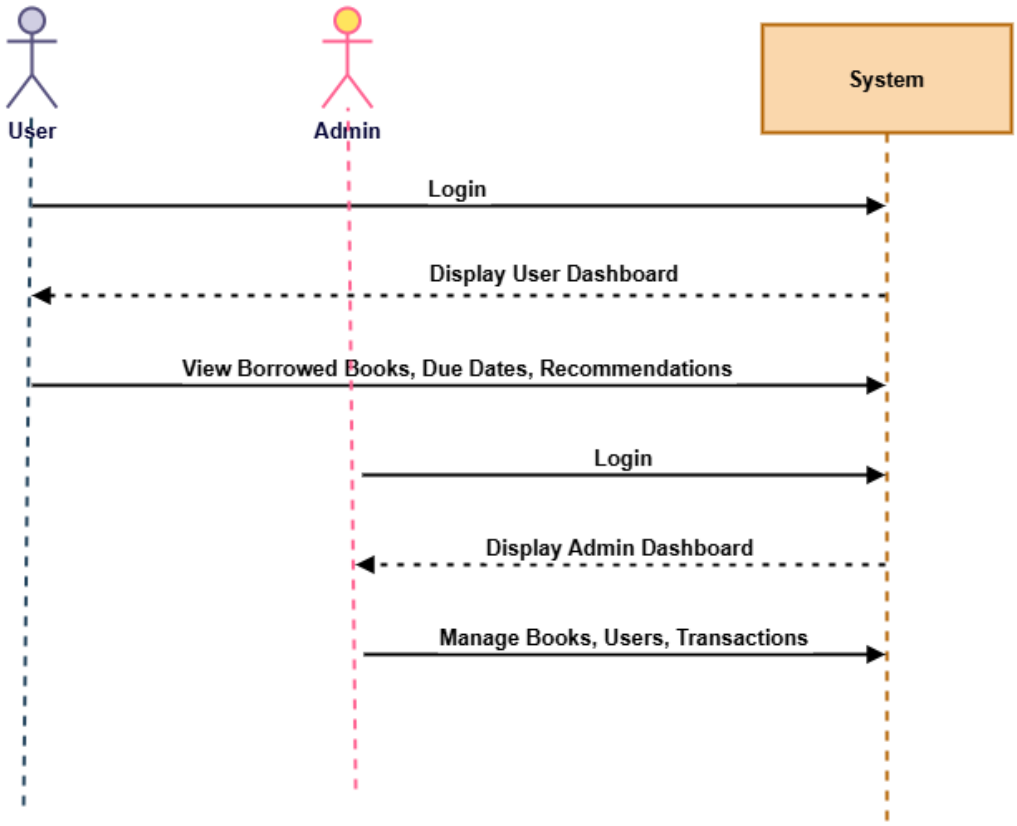

Figure 10 SD

The key process involves logging in to view personalized dashboards. The user logs in and views borrowed books, due dates, and recommendations. The admin logs in and manages books, users, and transactions. The system displays the appropriate dashboard based on the user's role and handles their actions accordingly.

c. Activity Diagram

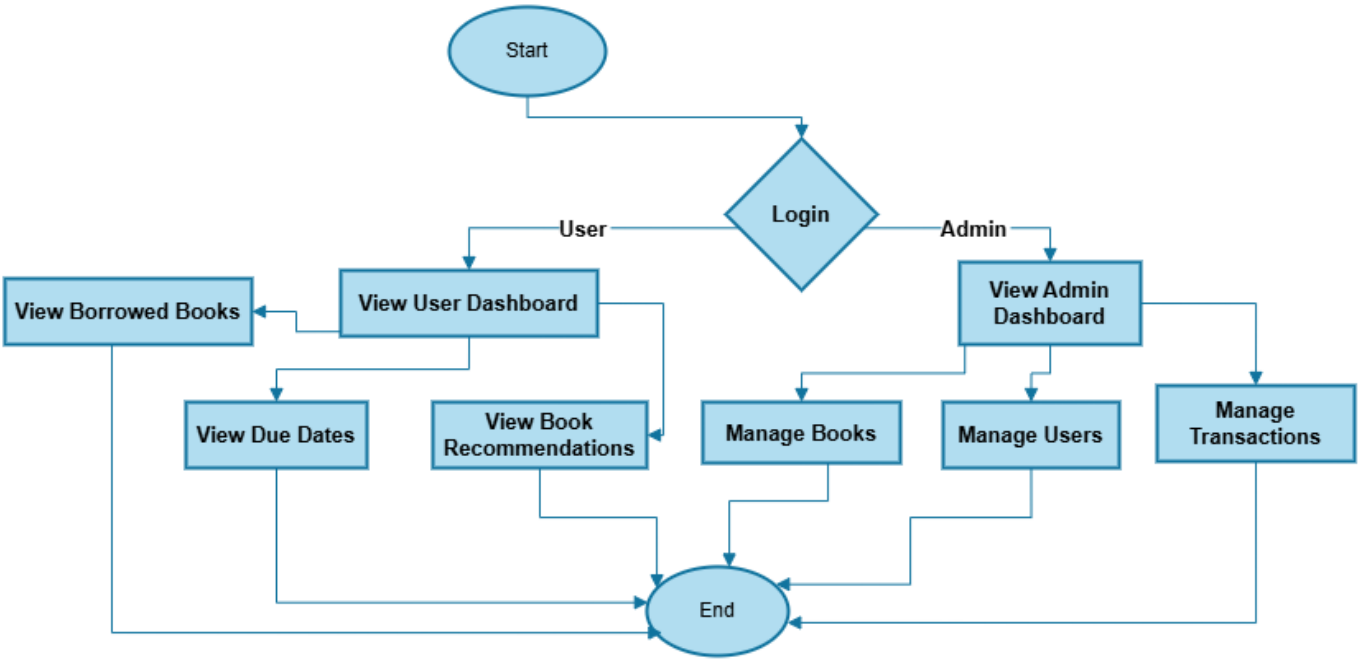

Figure 11 AD

This activity diagram shows how users and admins interact with personalized dashboards. Users can view borrowed books, due dates, and recommendations, while admins manage books, users, and transactions. The process starts with logging in and ends with viewing the respective dashboard information.

d. Wireframe

Low fidelity design of the user interface for this feature.

III. Recommendations Book

Use Case	Recommendations Book
Actors	User, System
Description	The system provides personalized book recommendations based on user preferences and borrowing history.

a. Detailed Use Case Diagram

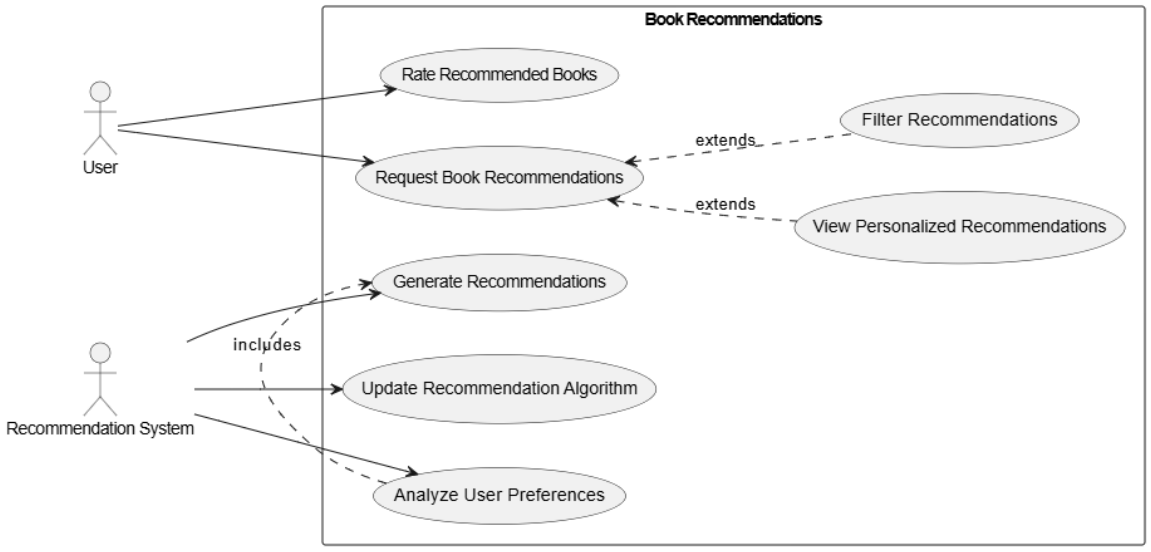

Figure 12 uc

This use case diagram shows how the User interacts with the book recommendation system. Users can request, rate, view, and filter recommendations. The system analyzes preferences, generates suggestions, and updates its algorithm. The diagram highlights key interactions and processes in a simple way.

b. Sequence Diagram

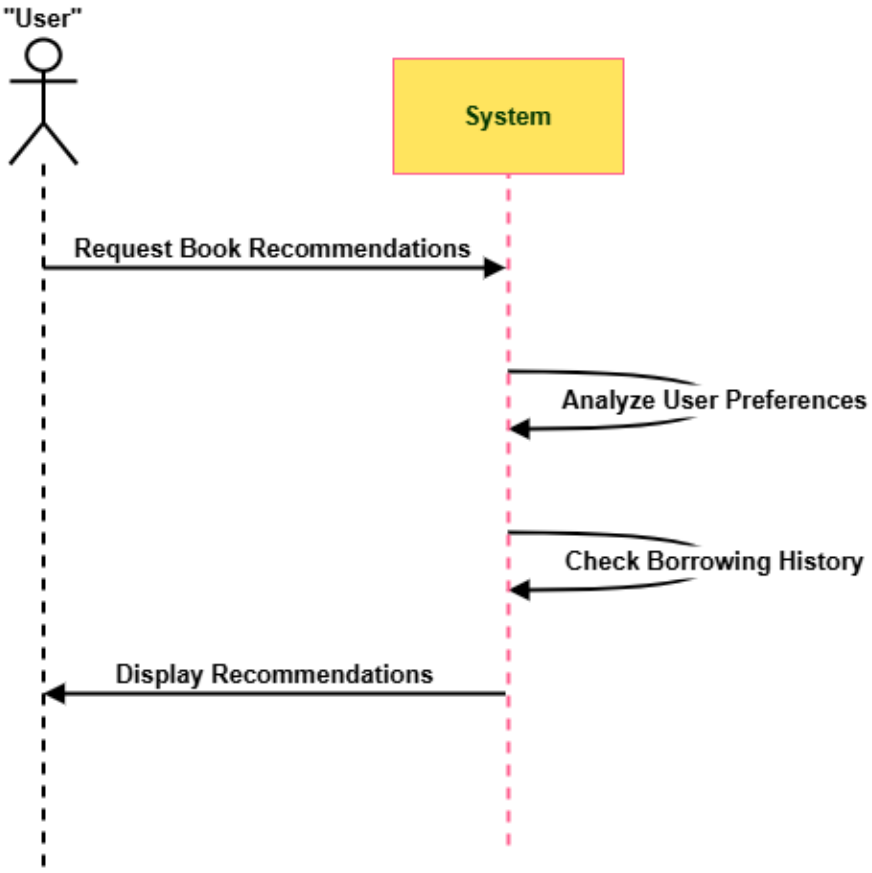

Figure 13 SD

The Sequence Diagram shows how the system handles book recommendations. The user requests recommendations, and the system analyzes their preferences and checks their borrowing history. Based on this information, the system displays personal recommendations to the user. This flow outlines how the system processes the request and responds with tailored suggestions.

c. Activity Diagram

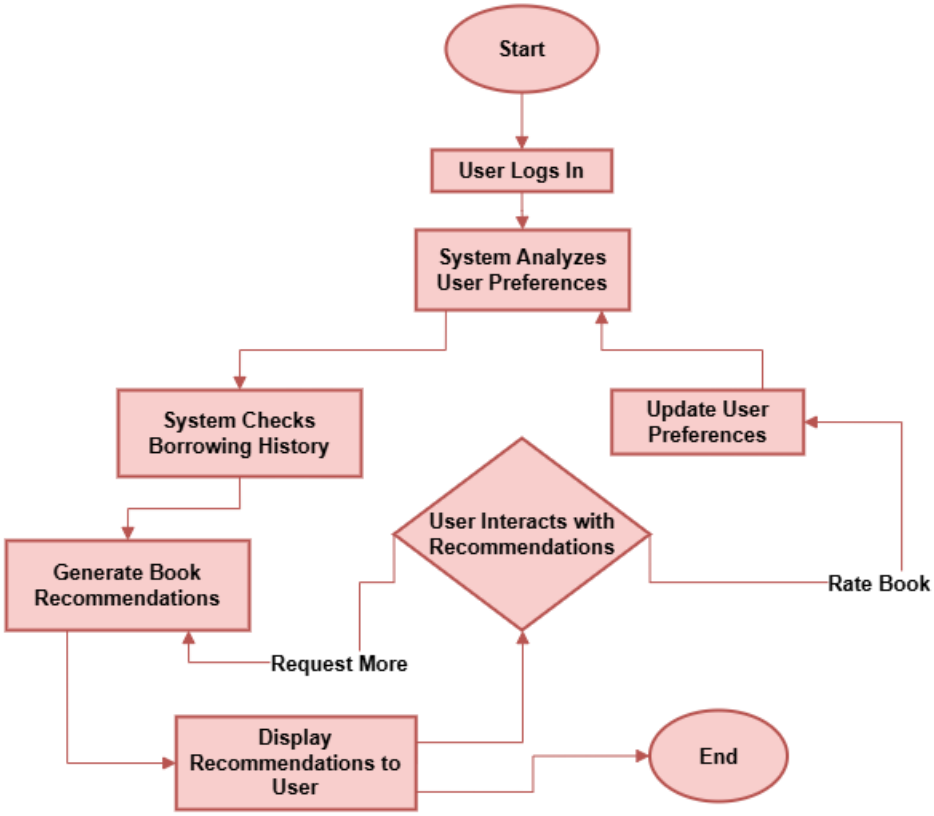

Figure 14 AD

The diagram shows the process of getting book recommendations. The user logs in, and the system checks their preferences and borrowing history. It then shows personalized recommendations. The user can ask for more or rate books, which updates future recommendations. The process ends after the user interacts with the recommendations.

d. Wireframe

Low fidelity design of the user interface for this feature.

IV. User Registration and Login

Use Case	User Registration and Login
Actors	User, System, Admin
Description	Users and admins can register or log in with their credentials; the system validates input, grants access based on roles, and shows errors if needed.

a. Detailed Use Case Diagram

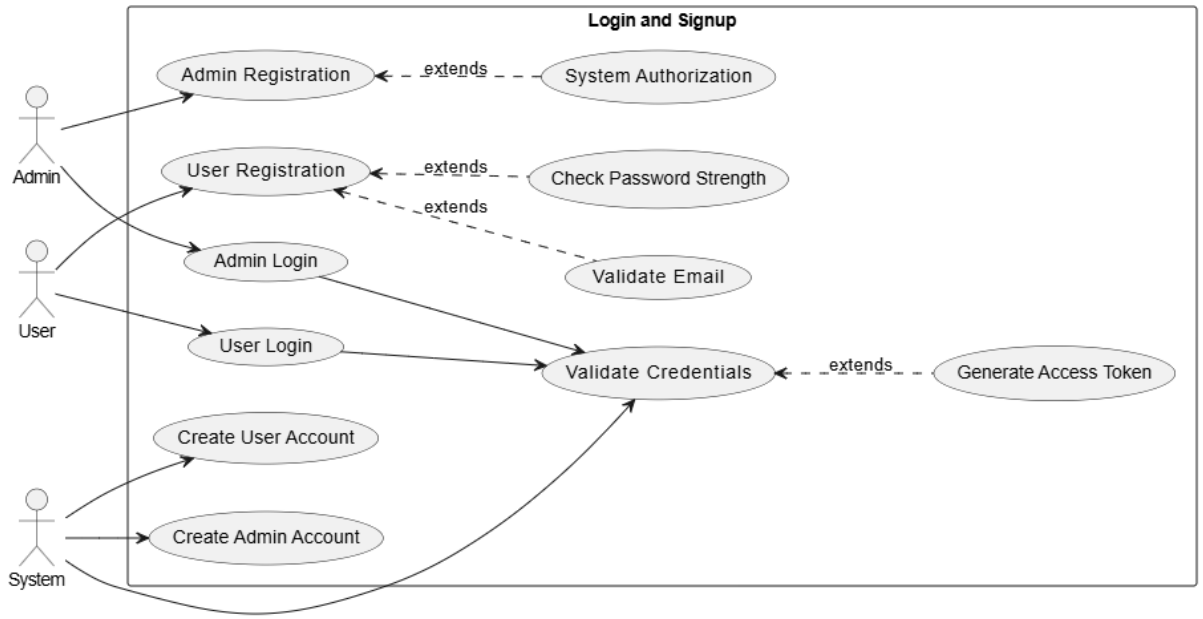

Figure 15 UC

The Use Case Diagram shows how users and admins interact with the system for login and signup. Users can register, log in, and have their credentials validated. Admins have a similar process but with additional privileges. The system handles account creation, credential validation, and access control. Extension points cover extra validation steps like email verification and password strength checks.

b. Sequence Diagram

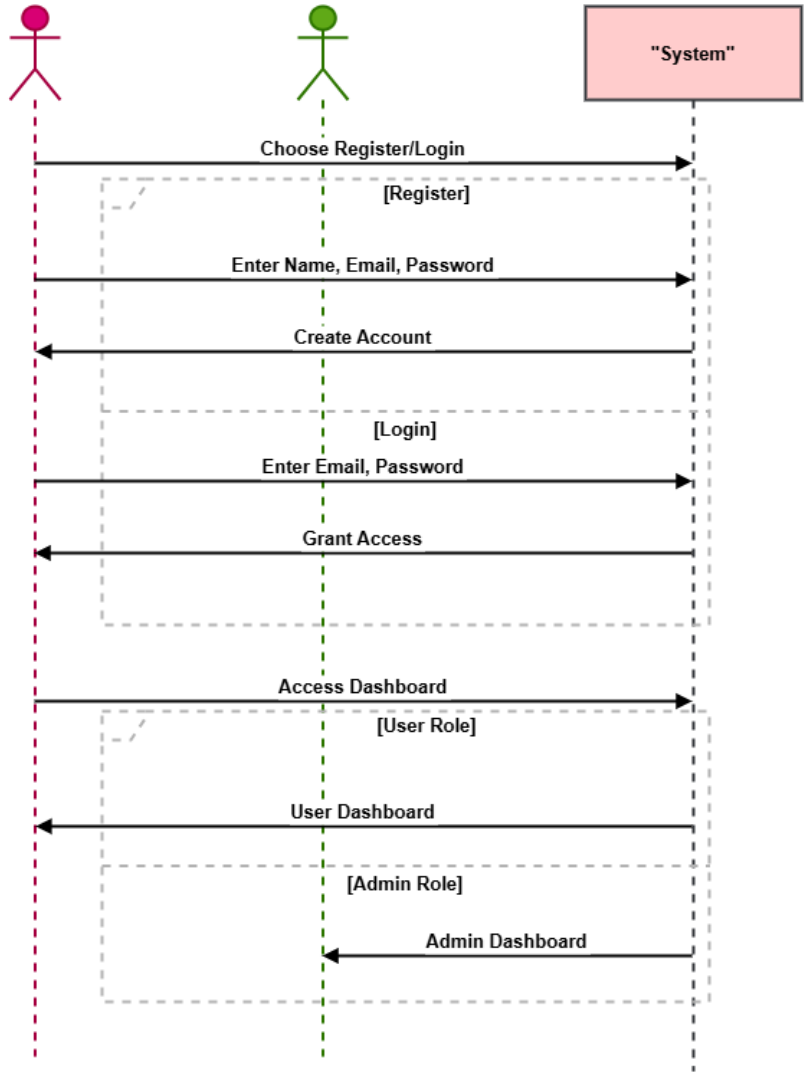

Figure 16 SD

The Sequence Diagram shows how the user or admin interacts with the system for registration or login. First, the user/admin chooses either "Register" or "Login." If registering, they enter their name, email, and password, and the system creates an account. If logging in, they enter their email and password, and the system grants access if the credentials are correct. After that, the user/admin can access their respective dashboard based on their role (User or Admin).

c. Activity Diagram

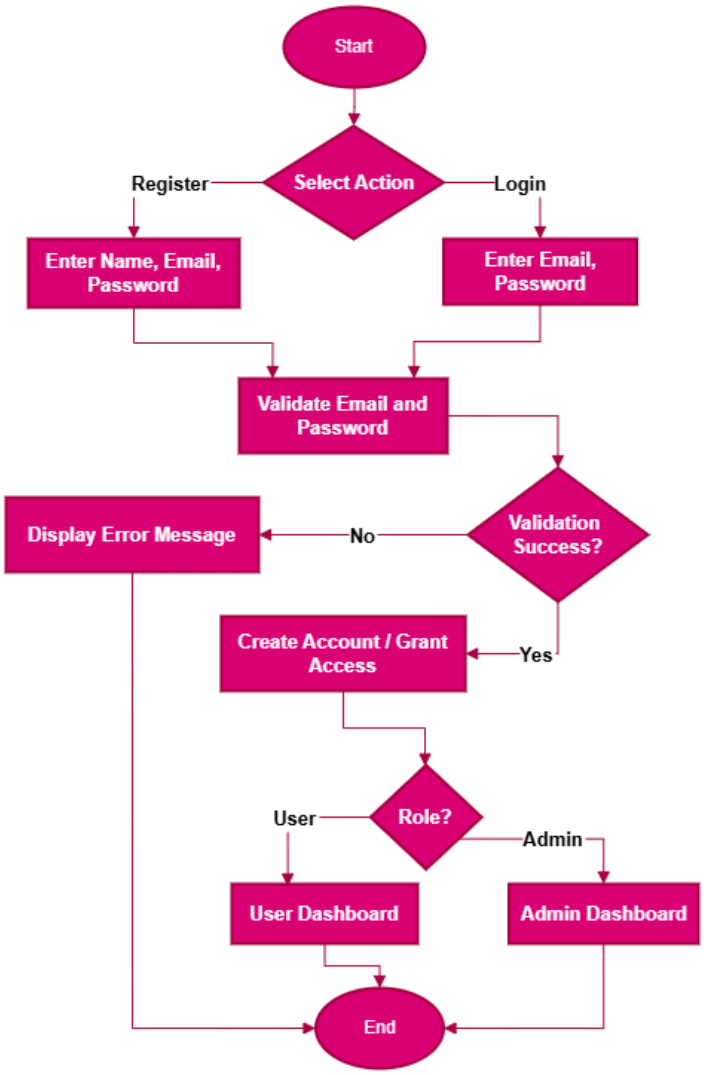

Figure 17 AD

The Activity Diagram shows the login and registration process. The user selects either "Register" or "Login." If registering, they enter their details, and the system validates them. If valid, an account is created. If logging in, the system verifies the credentials. If correct, access is granted based on the role (User or Admin). If invalid, an error message is shown. The process ends after the user reaches their dashboard or sees an error.

d. Wireframe

i) Automate Library Operations

Example:

Use Case	Automate Library Operations
Actors	User, Admin, System
Description	The system automates book borrowing, returning, and overdue fine calculations. Users can borrow books, and the system updates the records. If a book is overdue, the system calculates the fine.

e. Detailed Use Case Diagram

A detailed use case diagram showing specific scenarios for this feature.

f. *Sequence Diagram*

Diagram and explanation of the message flow for a key process in this feature.

g. *Activity Diagram*

Visual workflow for processes related to this feature.

h. *Wireframe*

Low fidelity design of the user interface for this feature.

4. Conclusion

The design of artefacts covered within this report has certainly been crucial in influencing the development phase of an automated library management system. Every diagram has offered vital information concerning the structure of the system, its interactions, and the flow of its work leading to a clear understanding of major functionalities. The Functional Decomposition Diagram had been a great benefit in breaking down the system into understandable components, whereas Use Case and ER diagrams clearly exposed user interactions and relationships within the database. Furthermore, a Class Diagram provided an object-oriented approach, whereas Activity Diagrams illustrated sequential workflows for several processes.

During the design phase, the following challenges were met integration between different parts of the system, optimization of database structures for the efficient retrieval of data.

They were all handled systematically through iterative design refinements, tested for validation under real-world scenarios. Hence, this phase of building the artifact can be stated as very strong in terms of system development, for it allows one to give clear guidance on how to implement the system. This phase will certainly illustrate the concrete representation of a system in software engineering; hence, it can use as a good reference for further enhancements in the system. More than helping in completing the present project, it gives a sound way of building a firm foundation for any subsequent software development project.

5. References

a) List all resources, tools, and references used for diagram creation.