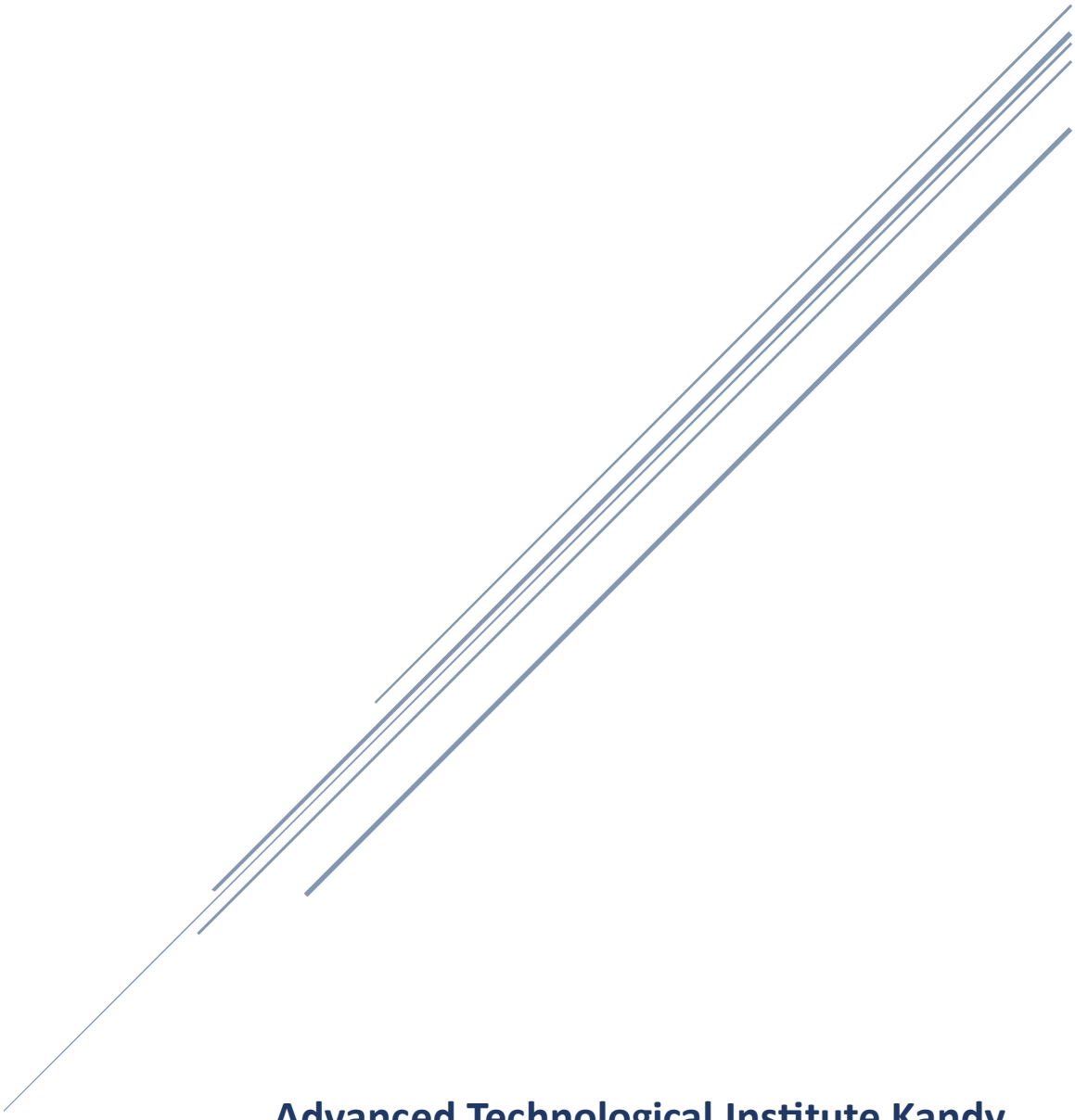
LIBRARY MANAGEMENT SYSTEM

KAN/IT/2022/F/0072
T.D Thumpalage



Advanced Technological Institute Kandy
Department of Information Technology

1. Introduction

The Library Management System (LMS) is a Java-based application designed to manage books and members in a library. It provides a graphical user interface (GUI) for easy interaction and adheres to the Model-View-Controller (MVC) architecture for organized code management. The system implements object-oriented programming (OOP) concepts such as encapsulation, inheritance, polymorphism, abstraction, and exception handling.

2. System Architecture

The LMS is developed using the MVC design pattern:

- **Model Layer:** Represents the data and business logic of the application. Classes include `Admin`, `Member`, `MembershipCard`, `Book`, and `Author`.
- **View Layer:** Contains GUI components to display the application's interface. Classes include `Login`, `EditBooks`, `EditMember`, and `View`.
- **Controller Layer:** Handles the interactions between the Model and View layers. Classes include `AdminController`, `BookController`, and `MemberController`.

The application structure ensures modularity, scalability, and ease of maintenance.

3. Object-Oriented Programming Concepts

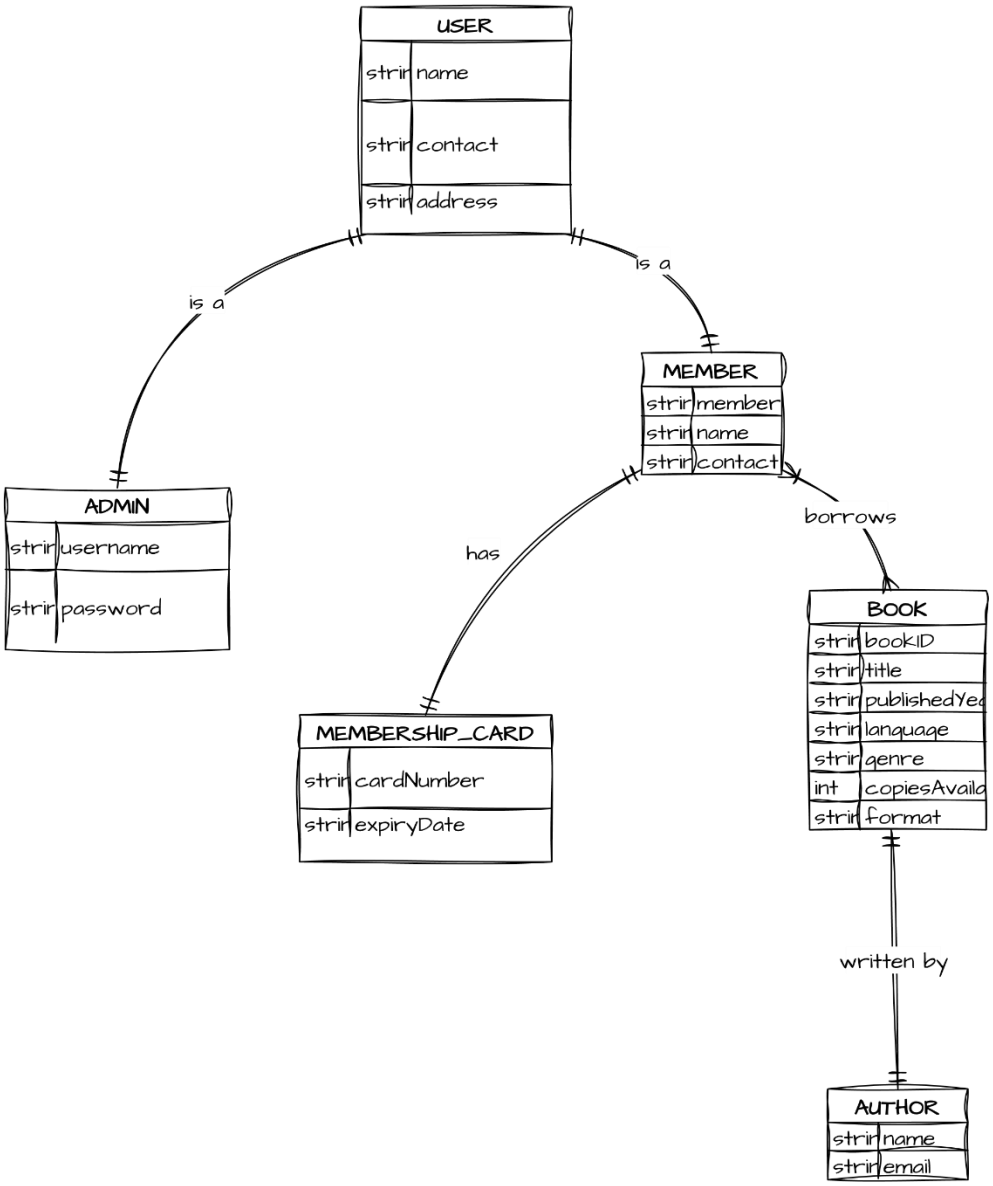
- **Encapsulation:** Private fields with public getters and setters are used in the `Book`, `Member`, and `MembershipCard` classes to protect data.
- **Inheritance:** The `User` class serves as a base class for `Admin` and `Member`.
- **Polymorphism:** The `displayDetails()` method is overridden in `Book` and `Member` classes to provide specific implementations.
- **Abstraction:** Interfaces or abstract classes (if used) define common behaviors.
- **Aggregation:** A `Book` is associated with multiple `Member` objects but exists independently.
- **Composition:** The `MembershipCard` class is part of the `Member` class, indicating a strong relationship.

4. Features and Functional Requirements

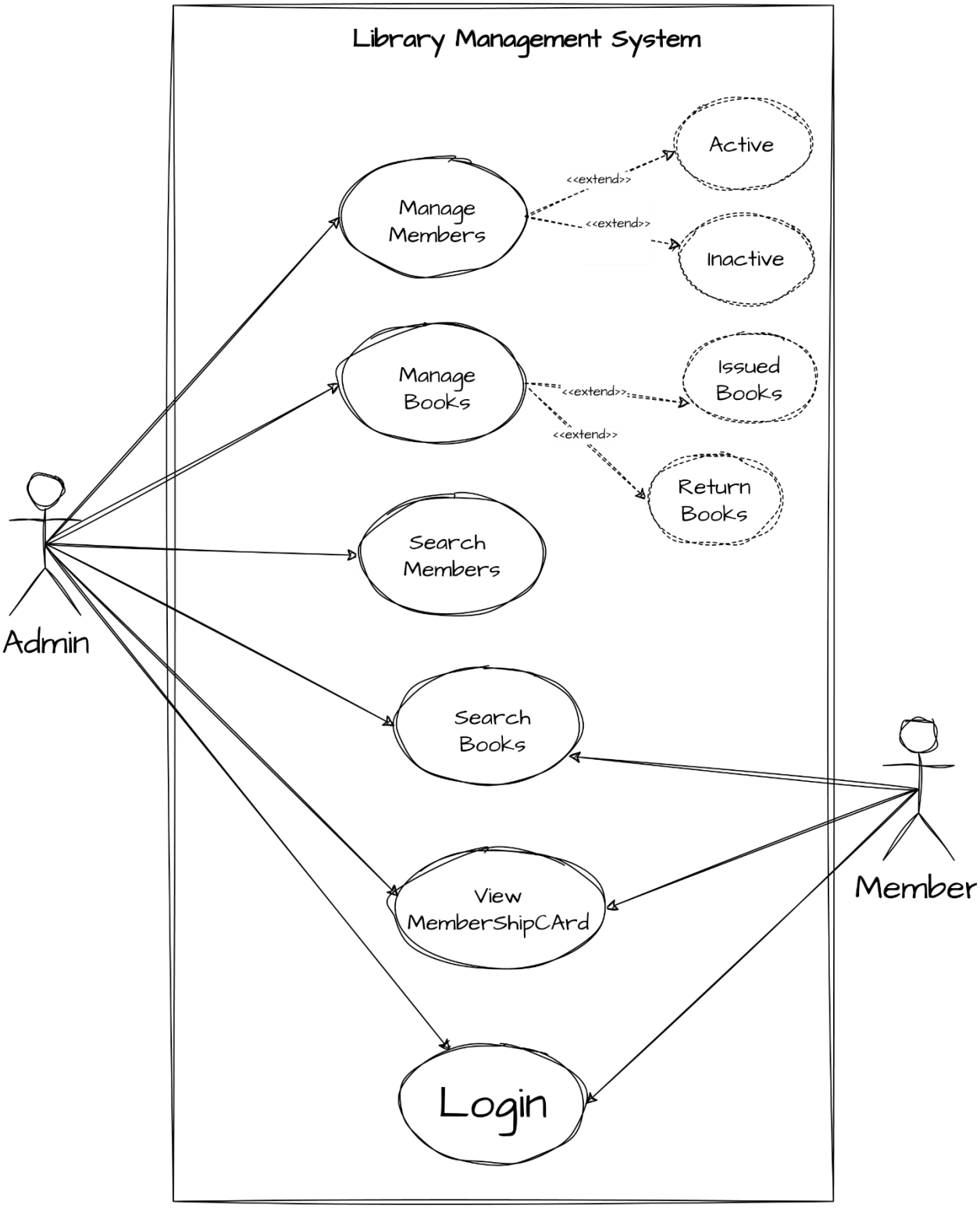
1. **Login System:**
    - Authenticates users as Admin or Member.
    - Provides different privileges based on roles.
2. **Member Management:**
    - Admin can add, update, delete, and view member details.
    - Each member has a `MembershipCard` containing details like card number and expiration date.

3. **Book Management:**
    - Admin can add, update, delete, and view book details.
    - Users can search books using overloaded methods:
        - `searchBooks(String title)`
        - `searchBooks(String title, String author)`
4. **Exception Handling:**
    - Handles invalid login attempts, empty form submissions, and unauthorized access.

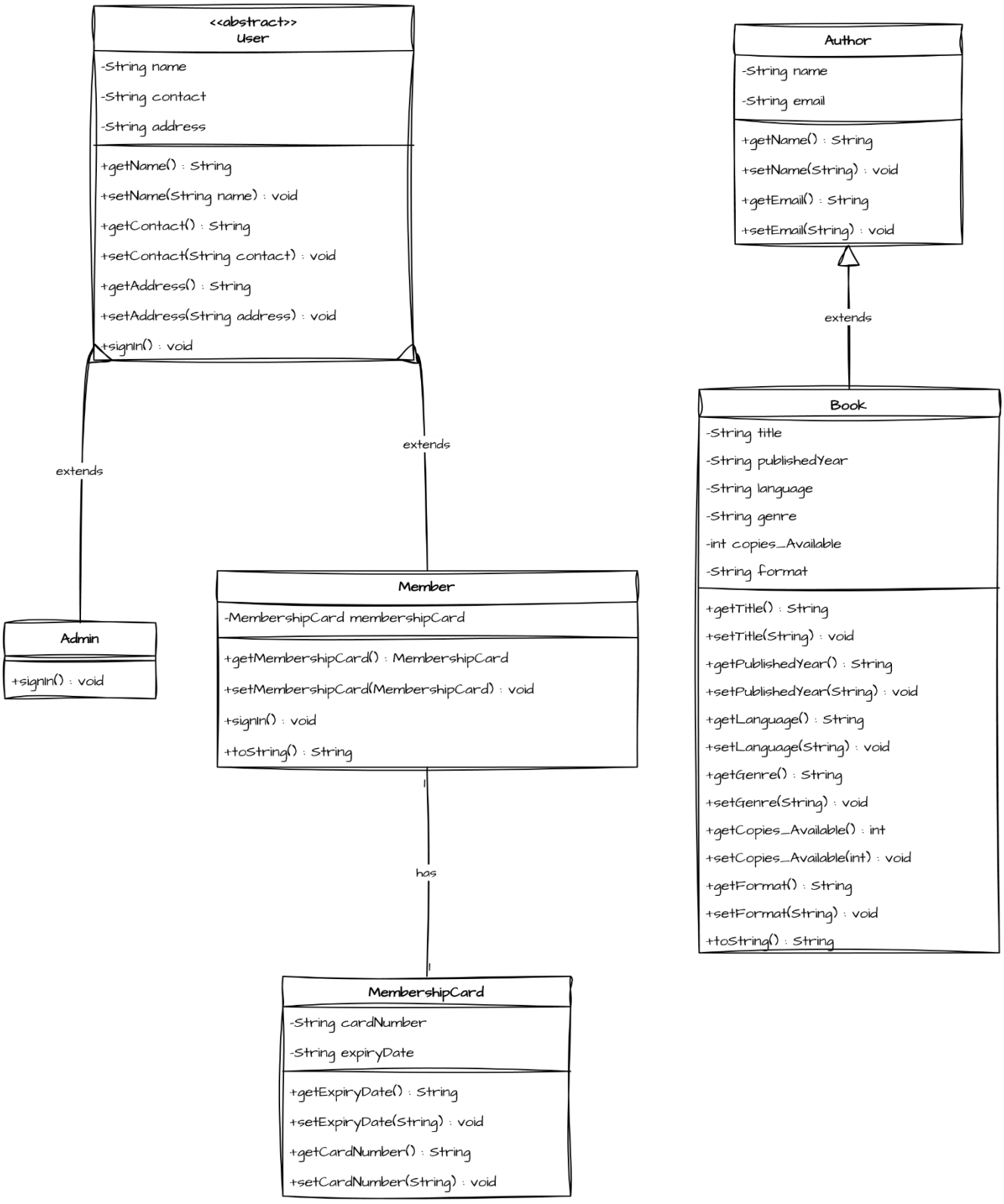
5. Entity-Relationship Diagram



6. Use Case Diagram



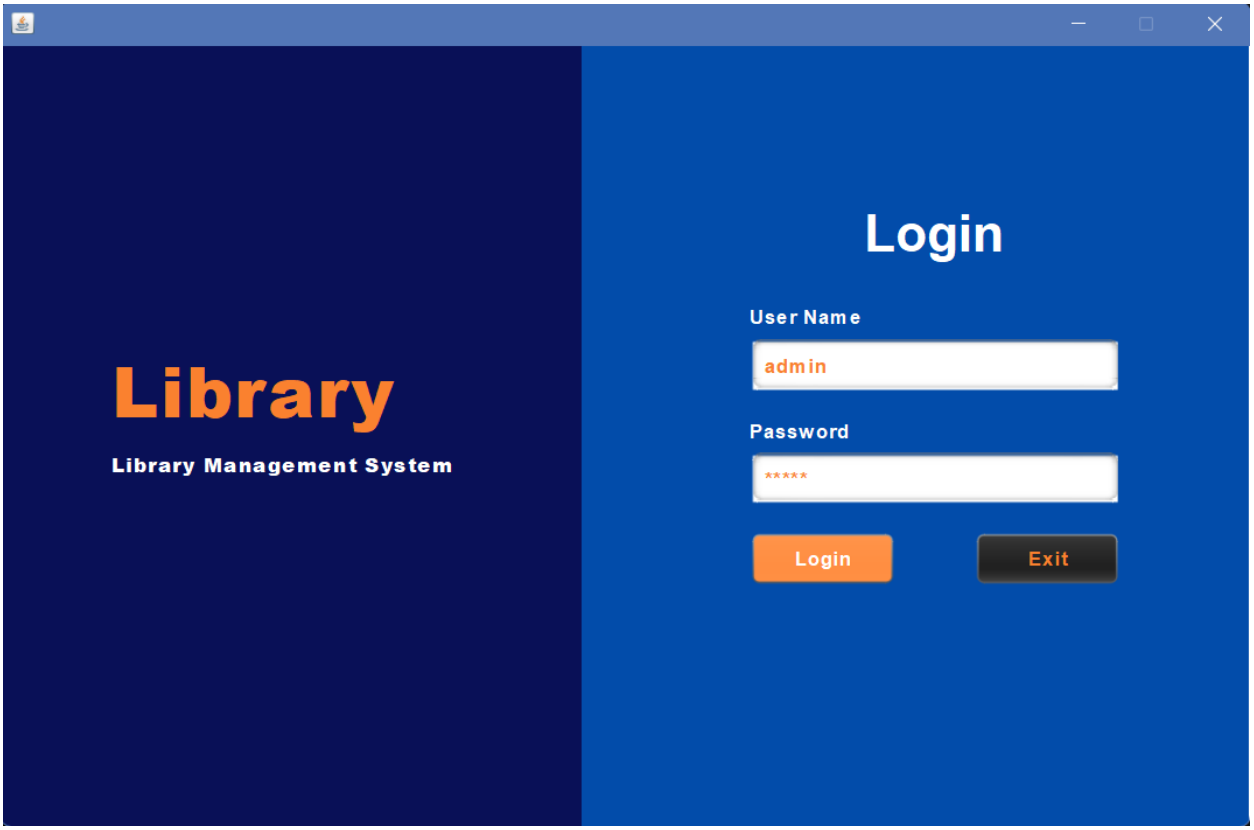
7. Class Diagram



8. User Manual

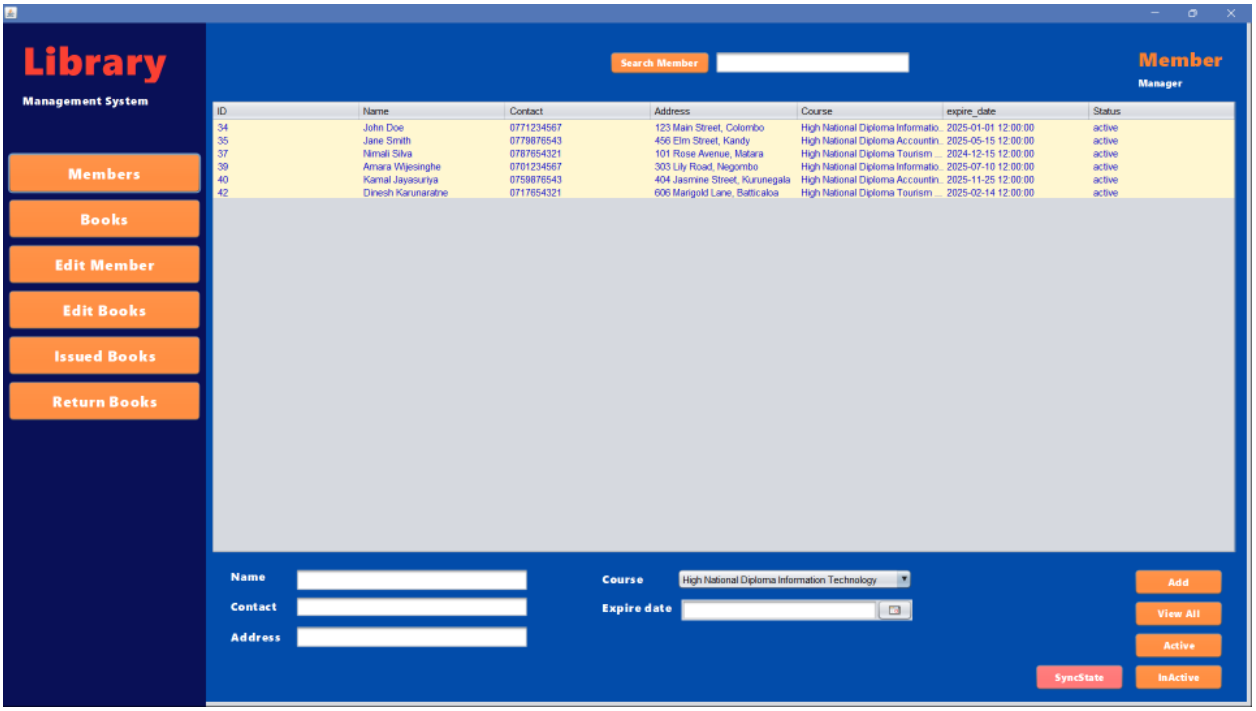
Login

- Launch the application.
- Enter the username and password.
- Click "Login." Admin users access full features, while Members access limited features.



9. Member Management:

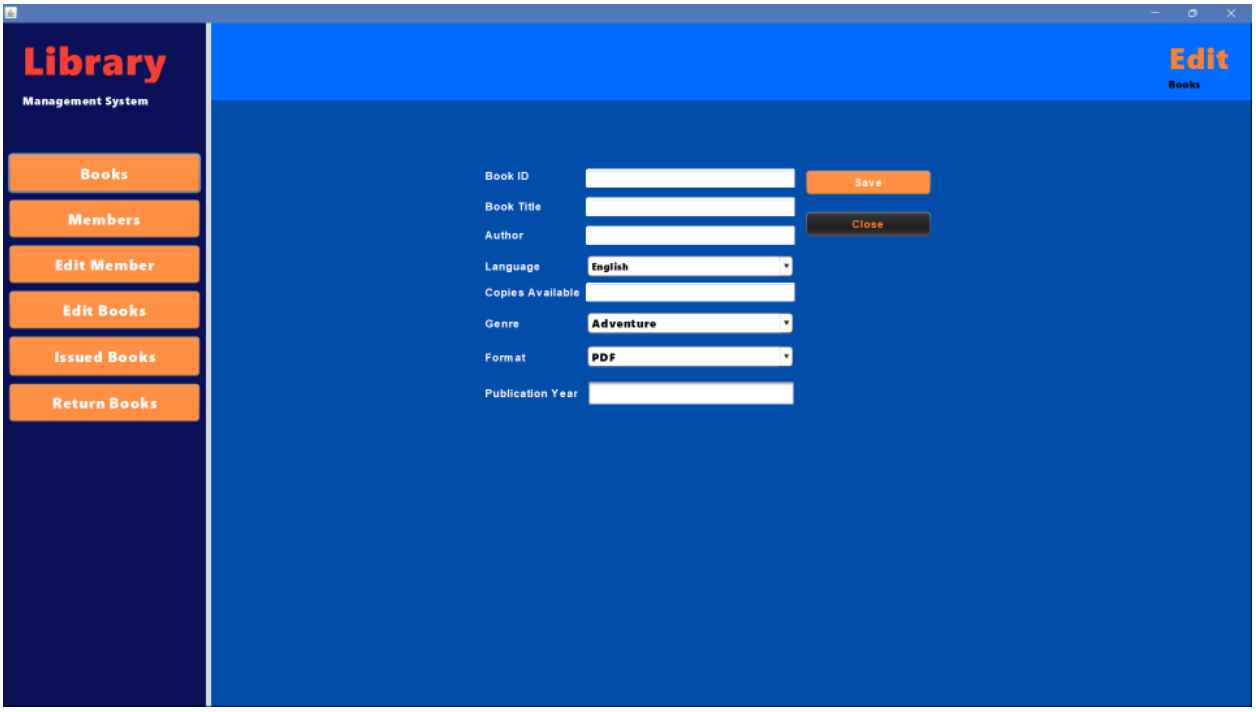
1. Navigate to "Manage Members" (Admin only).
2. Use the "Add," "Edit," or "Active" or “InActive” buttons to perform actions.



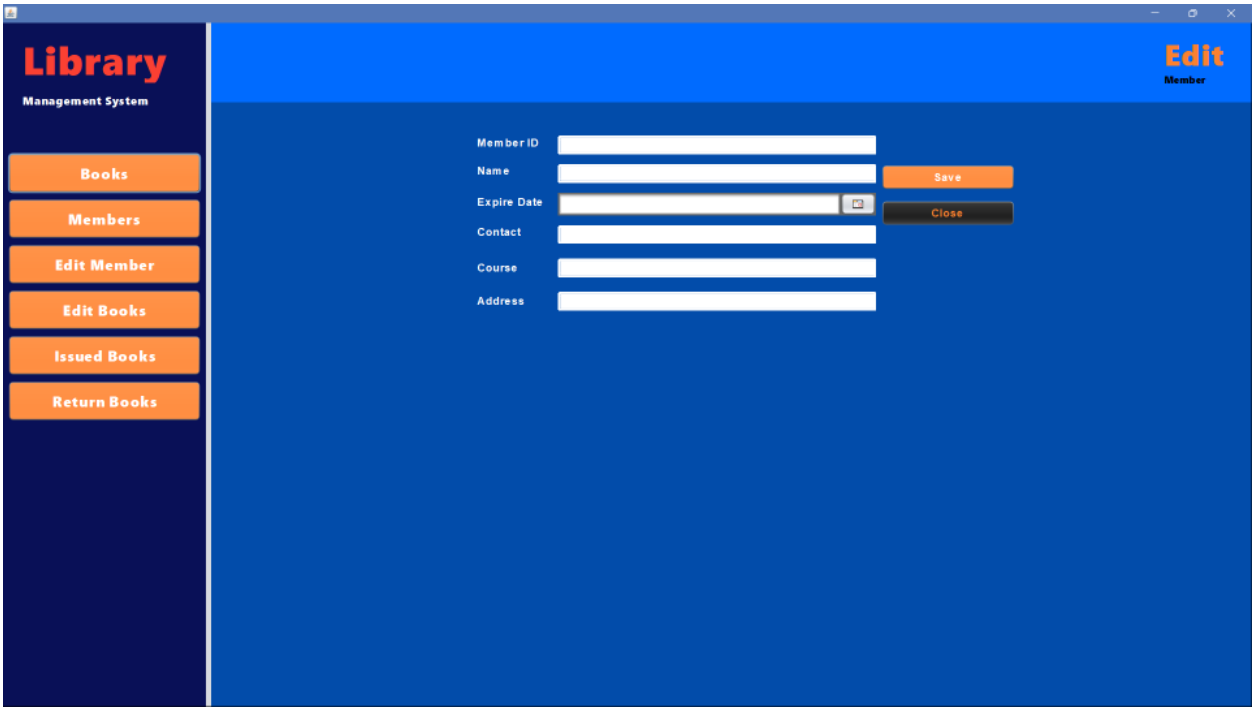
10. Book Management:

1. Navigate to "Manage Books" (Admin only).
2. Search books by title or title and author.

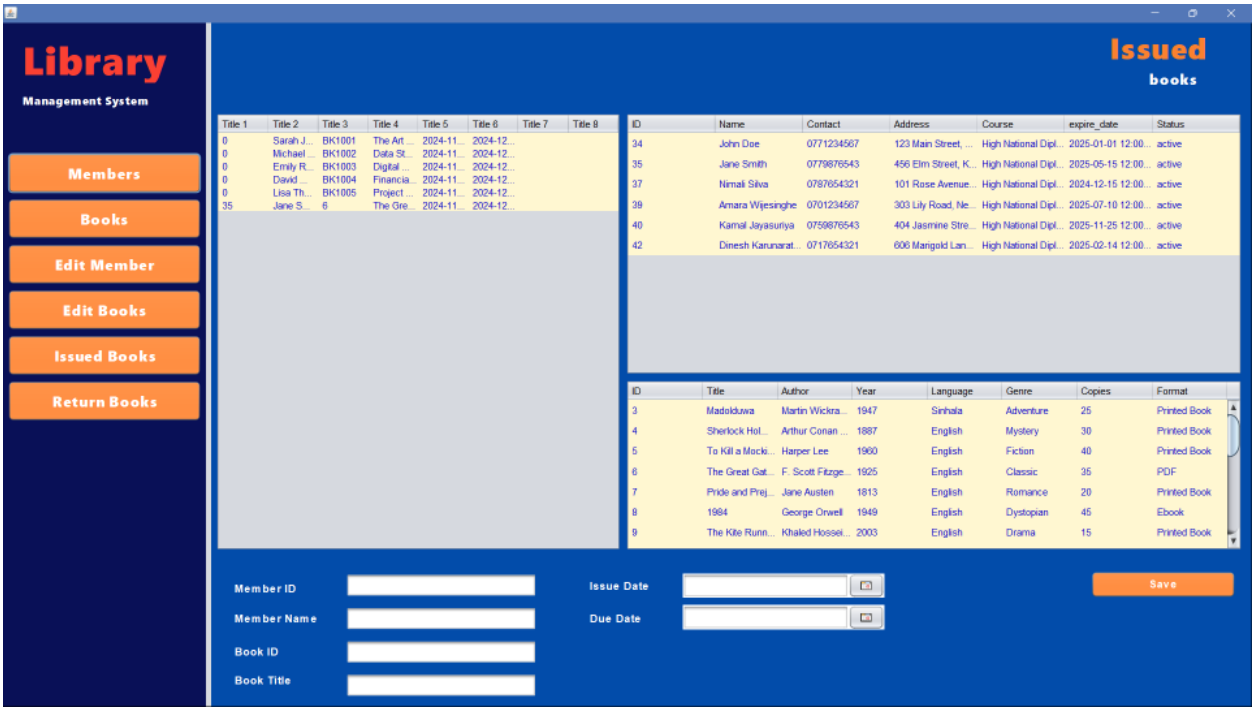
11. Edit Book:



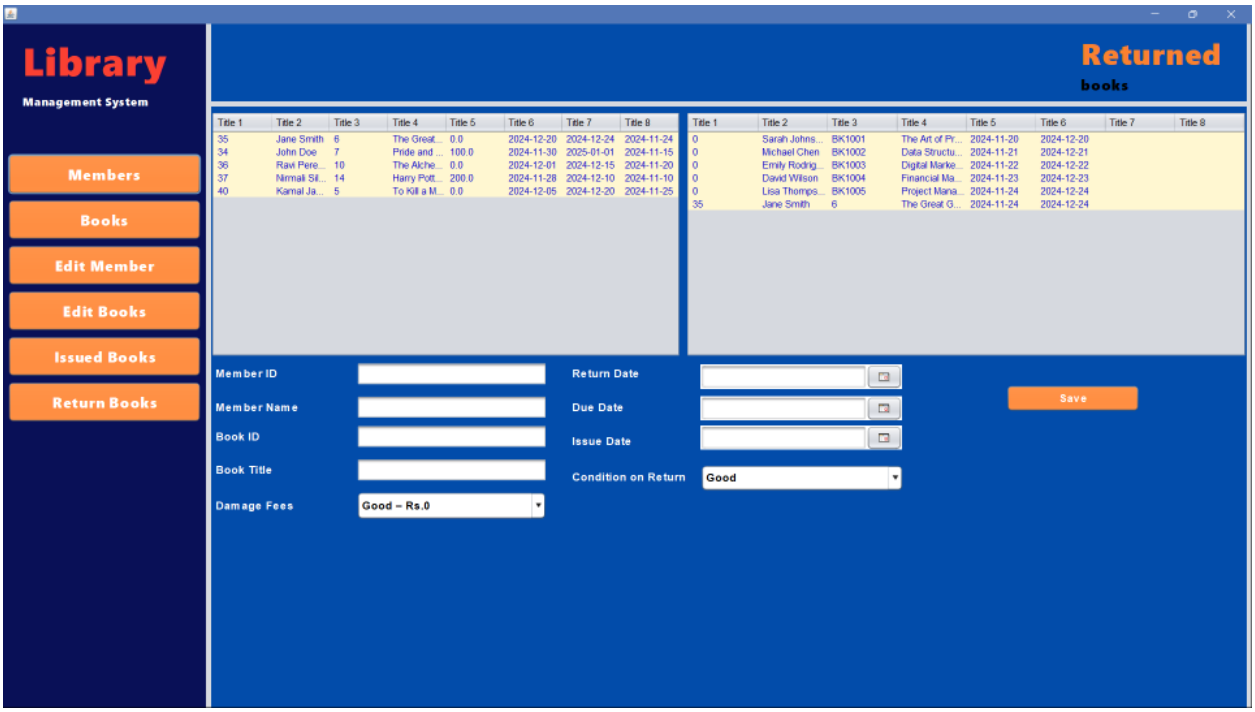
12. Edit Member:



13. Issued Books:



14. Issued Books:



15. Conclusion

The LMS provides a robust and efficient way to manage library operations, ensuring user-friendly interfaces and secure access. Future enhancements may include integrating a database for persistence and expanding member privileges.