1️⃣ ER Diagram (Entity Relationship Diagram)

Entities and Attributes:

1. Book

Book_ID (PK)

Title

Author

Publisher

Category

ISBN

Quantity

Available_Copies

2. Member

Member_ID (PK)

Name

Department

Email

Contact_No

Address

- **3. Issue**
- Issue_ID (PK)
- Book_ID (FK)
- Member_ID (FK)
- Issue_Date
- Due_Date
- Return_Date
- Fine
- **4. Librarian**
- Librarian_ID (PK)
- Name
- Username
- Password
- Contact
- **Relationships:**
- **Member → Issue (1-to-M)** (A member can issue many books)
- **Book → Issue (1-to-M)** (A book can be issued multiple times)
- **Librarian → Issue (1-to-M)** (A librarian handles many issue records)

- **2️⃣ Use Case Diagram**

- **Actors:**
- **Admin / Librarian**
- **Member (Student / Faculty)**
- **Use Cases:**
- Login / Logout
- Add / Update / Delete Books
- Register Member
- Issue Book
- Return Book
- Calculate Fine
- Generate Report
- Search Books
- **Relationships:**
- Librarian → manages all use cases except “Search Books”.
- Member → can “Login”, “Search Books”, “View Issued Books”, “Request Issue / Return”.

3️⃣ Sequence Diagram

- **Scenario: Issue Book Process**
- **Actors:** Member, Librarian, System, Database
- **Steps:**
- Member requests a book to Librarian.
- Librarian searches the book in the system.
- System queries the database for book availability.
- If available → Librarian enters Member_ID and Book_ID.
- System records Issue_Date and Due_Date.
- System updates book availability in database.
- System displays confirmation message.

- 4️⃣ **Activity Diagram**

- **Process: Borrowing a Book**
- **Activities:**
- Login
- Search for Book
- Check Book Availability
- If available → Fill Issue Form
- Save Issue Record
- Update Book Quantity
- Confirmation to User
- Logout
- **Decision Points:**
- If *Book not available* → Display “Unavailable” and end.
- If *Book available* → Proceed to issue process.

5️⃣ Class Diagram

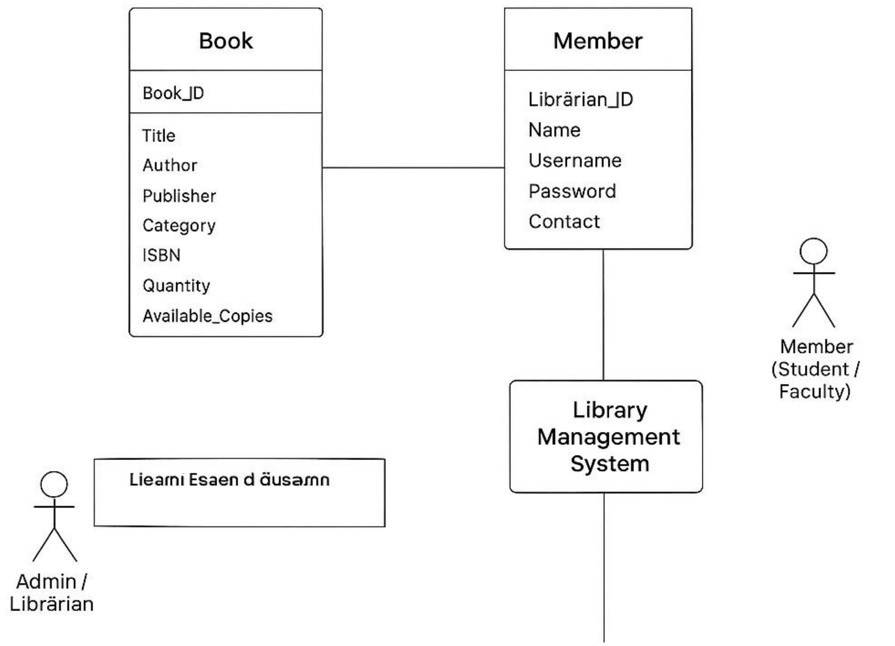
- **Classes and Attributes:**
- **Class: Book**
- bookID
- title
- author
- category
- quantity
- **Methods:** addBook(), updateBook(), deleteBook(), searchBook()
-

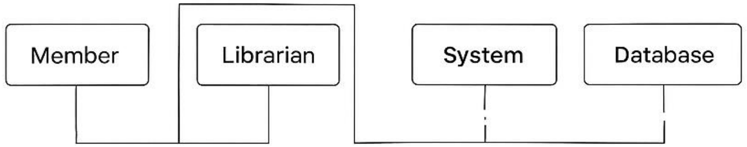
- **Class: Member**
- memberID
- name
- email
- department
- **Methods:** register(), updateInfo(), viewIssuedBooks()

- **Class: Librarian**
- librarianID
- name
- username
- password
- **Methods:** login(), manageBooks(), issueBook(), returnBook()
-

- **Class: Issue**
- issueID
- issueDate
- dueDate
- returnDate
- fine
- **Methods:** calculateFine(), updateReturnDate()
-

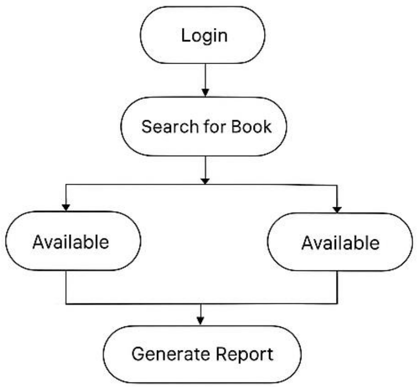
- **Relationships:**
- Librarian *manages* Book, Member, Issue
- Member *borrows* Book through Issue
- Book *associated* with Issue (1-to-M)

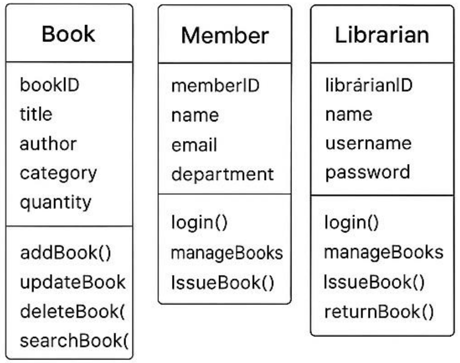




Issue Book

Acmivity Diagram





Class Diagram