Define Functional and Non-Functional Requirements

functional Requirements

User Management

User Accounts - enroll users, sign them in or out and control the account of librarians and users.
Security precaution that can be applied for the classification of staff (for example, librarians) or patrons.

Book Management

Browse, create, modify, and remove books.
Monitor the books' stock including borrowed status.

Borrowing and Returning Books:

Borrow books and also return the books that one had borrowed.
Record the due dates of books that are due and books that have not been returned yet.
Inform users that their books are due and inform users that their requested books are available for pick up.

Catalog Search

Research novels by their name, the name of the author and type of fiction or the ISBN number.
It is necessary to select only available records by using filters.

Reservation System

Let the users reserve books.
Inform users when the books that they have reserved are available.

Reporting

Create reports on books that have been borrowed, books that are overdue, and activity of the users.

Non-Functional Requirements

Performance

Maximum number of users that system should be able to accommodate is equal to 500.
These tasks at News time should take under 2 seconds of search to return the results.
Security:

Sometimes, the data can be encrypted to help in protecting the sensitive information that the data contains.
The first important technology requirement concerns the achieving of secure user authentication and authorization.

Usability

Ease of use: This concept relates both to the librarians and the users of the/library services it offers.
For any device layout adaptation.

Scalability:

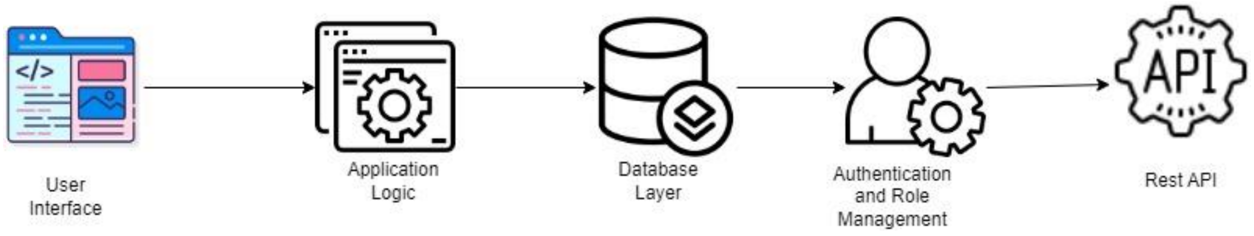
System has to be expandable in a way that it will be able to accommodate more books and more users in the future.

Reliability

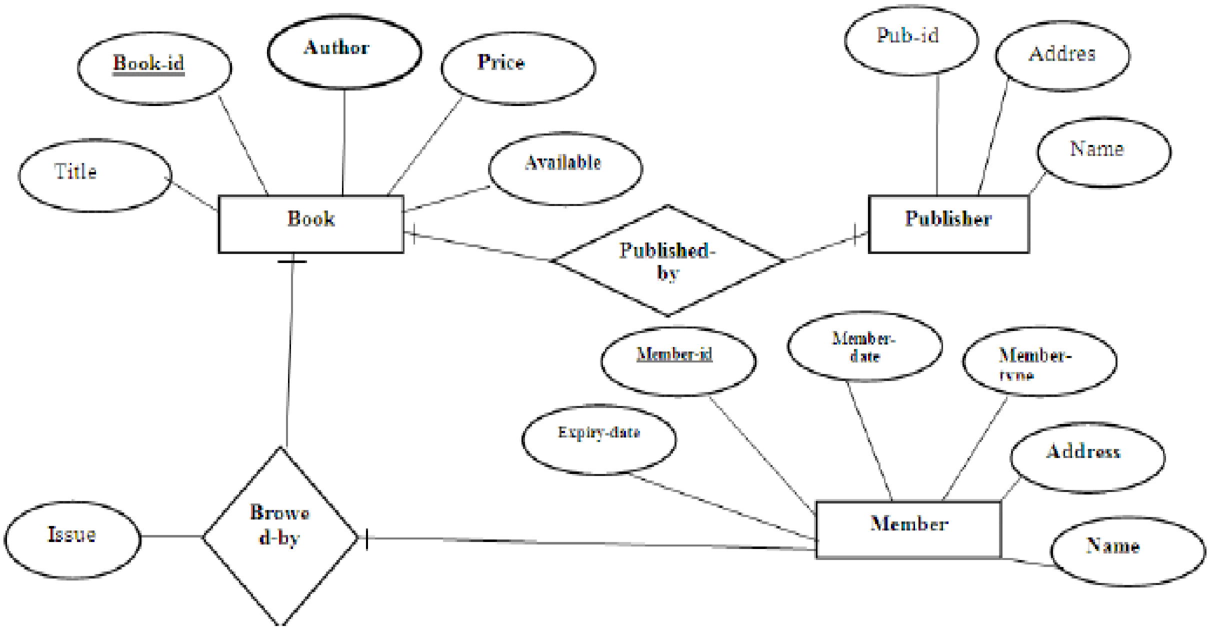
The system must have high availability and as low possible downtime as possible.

Architecture Diagram

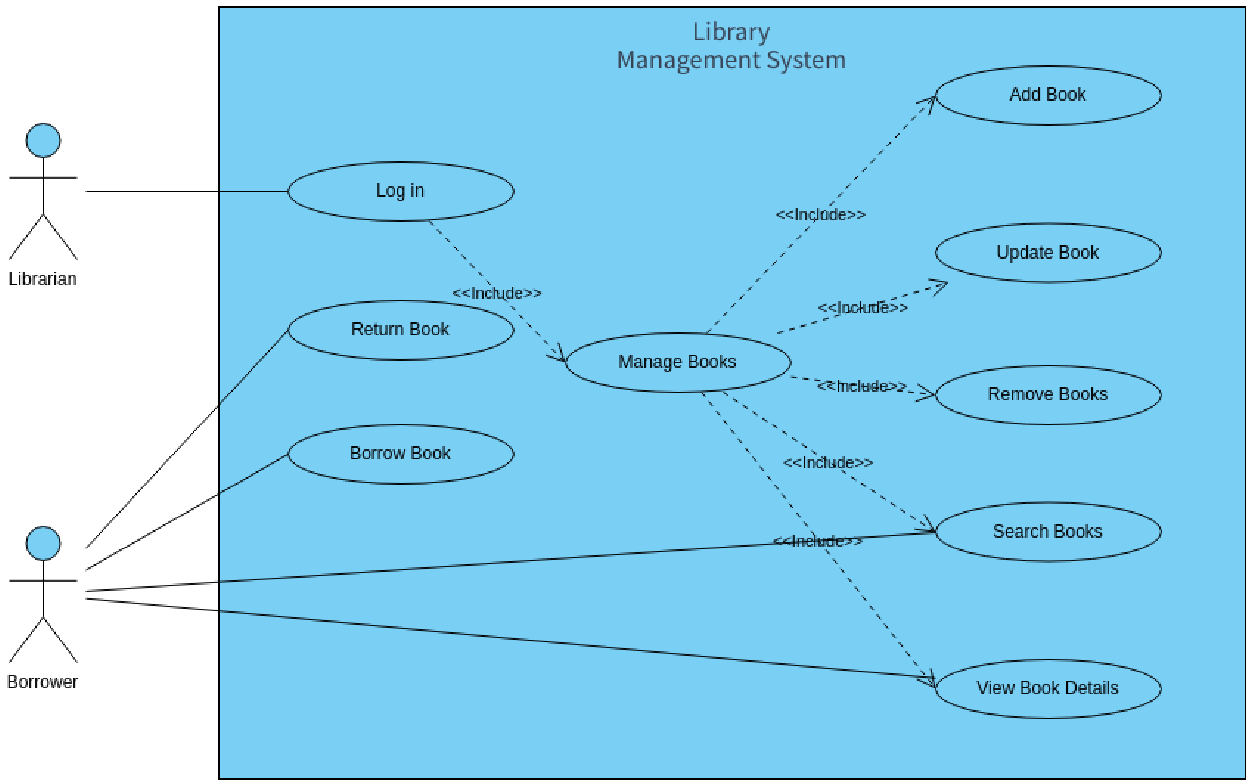
- **High-Level Architecture Diagram:**
  - **User Interface/ Frontend:** Laravel Blade templates.
  - **Backend:** Laravel application handling business logic.
  - **Database:** MySQL for data storage.
  - **Server:** Web server (Apache/Nginx) and Laravel running on it.
  - **Rest Apis:** Provides API endpoints for interacting with the application.



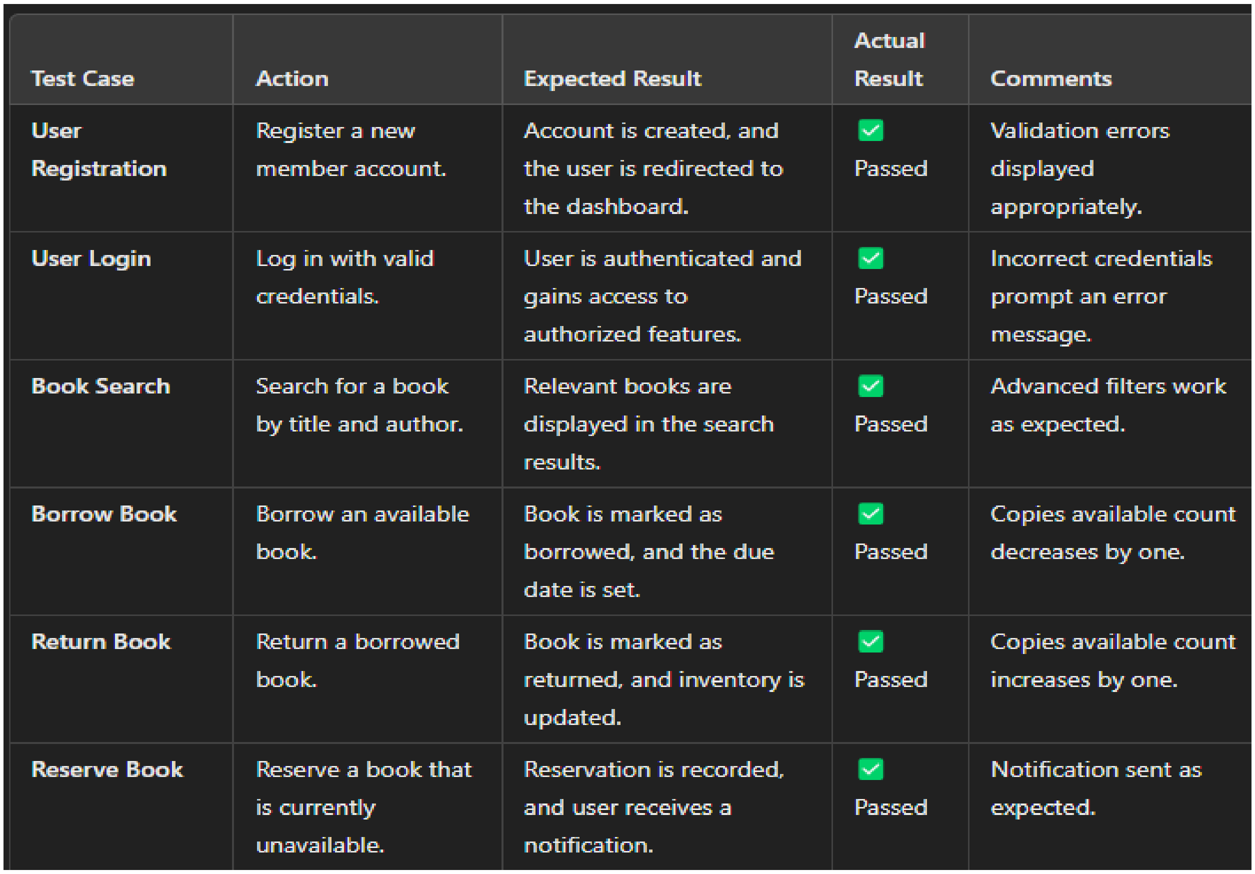
ER Diagram



Use Case Diagram



Test Cases



Test Case	Action	Expected Result	Actual Result	Comments
User Registration	Register a new member account.	Account is created, and the user is redirected to the dashboard.	✅ Passed	Validation errors displayed appropriately.
User Login	Log in with valid credentials.	User is authenticated and gains access to authorized features.	✅ Passed	Incorrect credentials prompt an error message.
Book Search	Search for a book by title and author.	Relevant books are displayed in the search results.	✅ Passed	Advanced filters work as expected.
Borrow Book	Borrow an available book.	Book is marked as borrowed, and the due date is set.	✅ Passed	Copies available count decreases by one.
Return Book	Return a borrowed book.	Book is marked as returned, and inventory is updated.	✅ Passed	Copies available count increases by one.
Reserve Book	Reserve a book that is currently unavailable.	Reservation is recorded, and user receives a notification.	✅ Passed	Notification sent as expected.