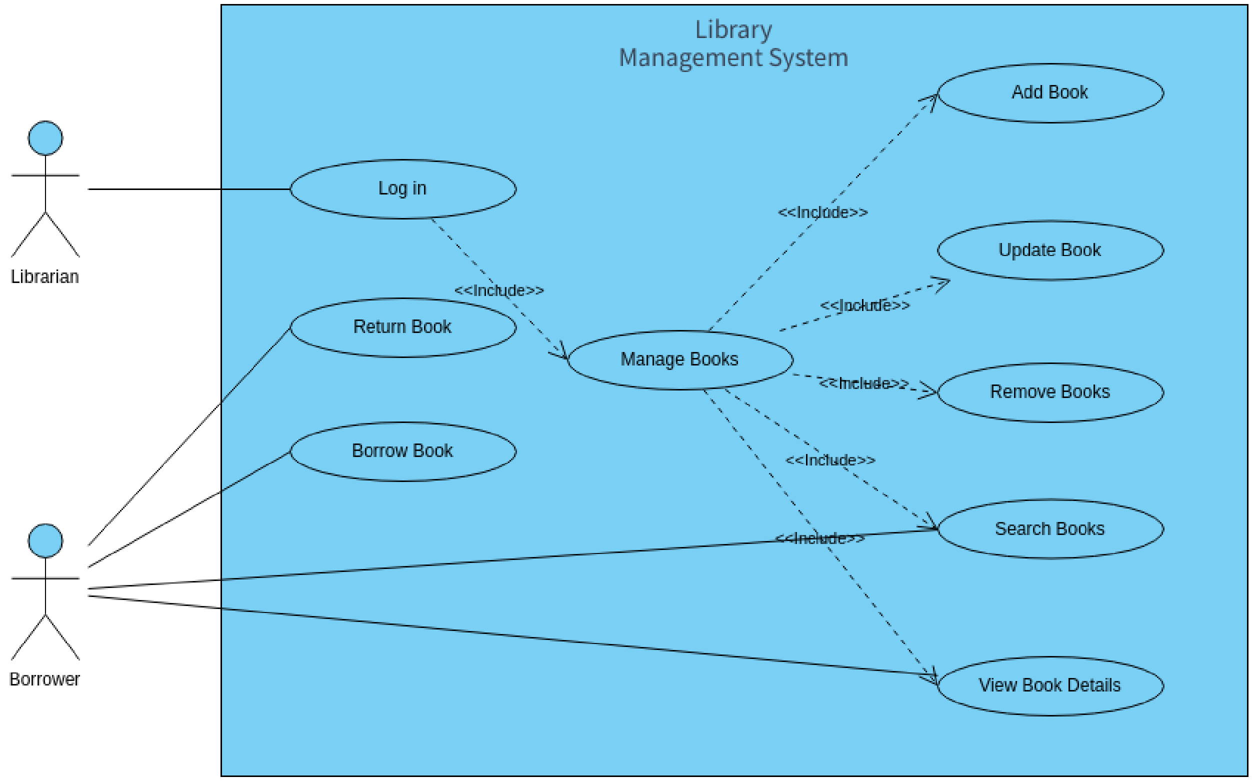
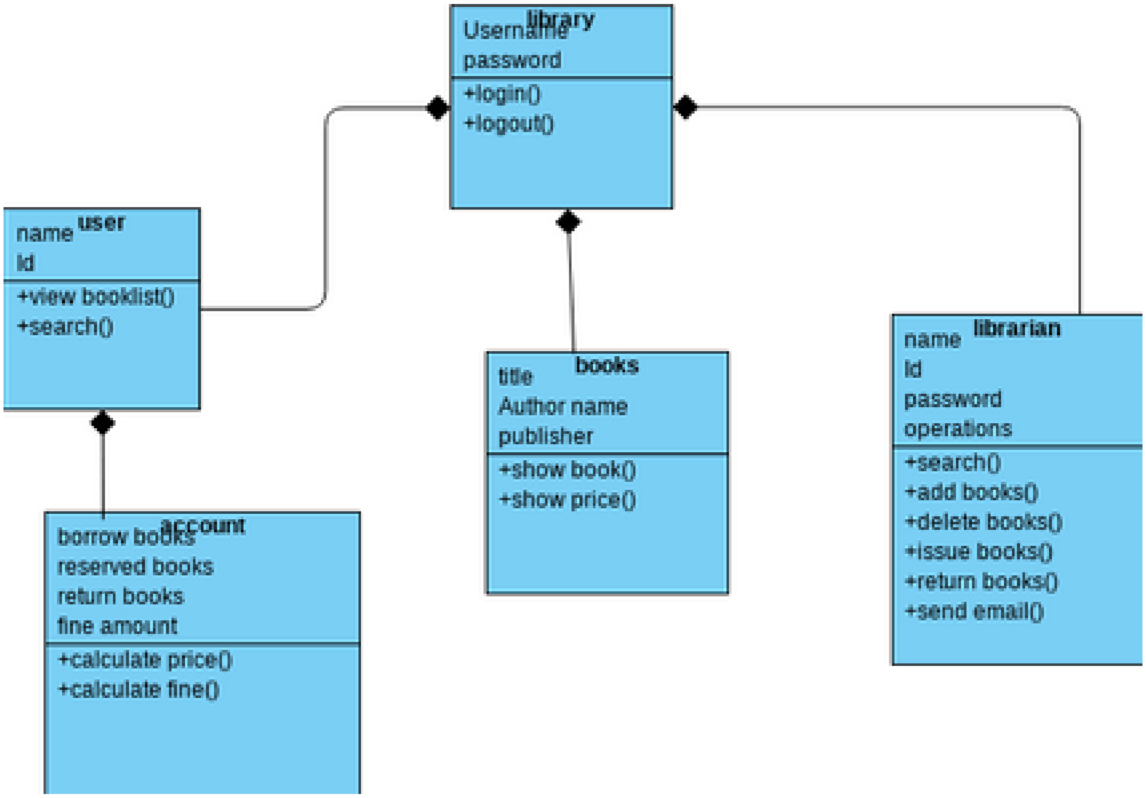
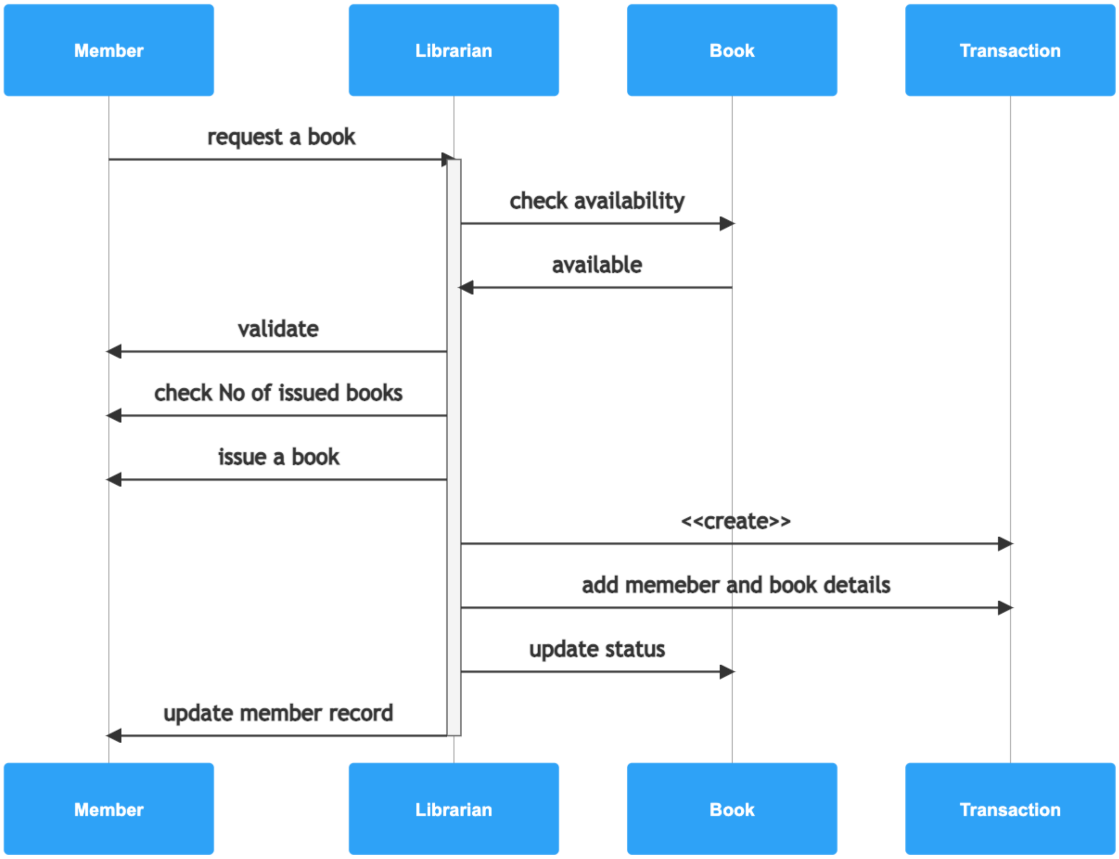
library

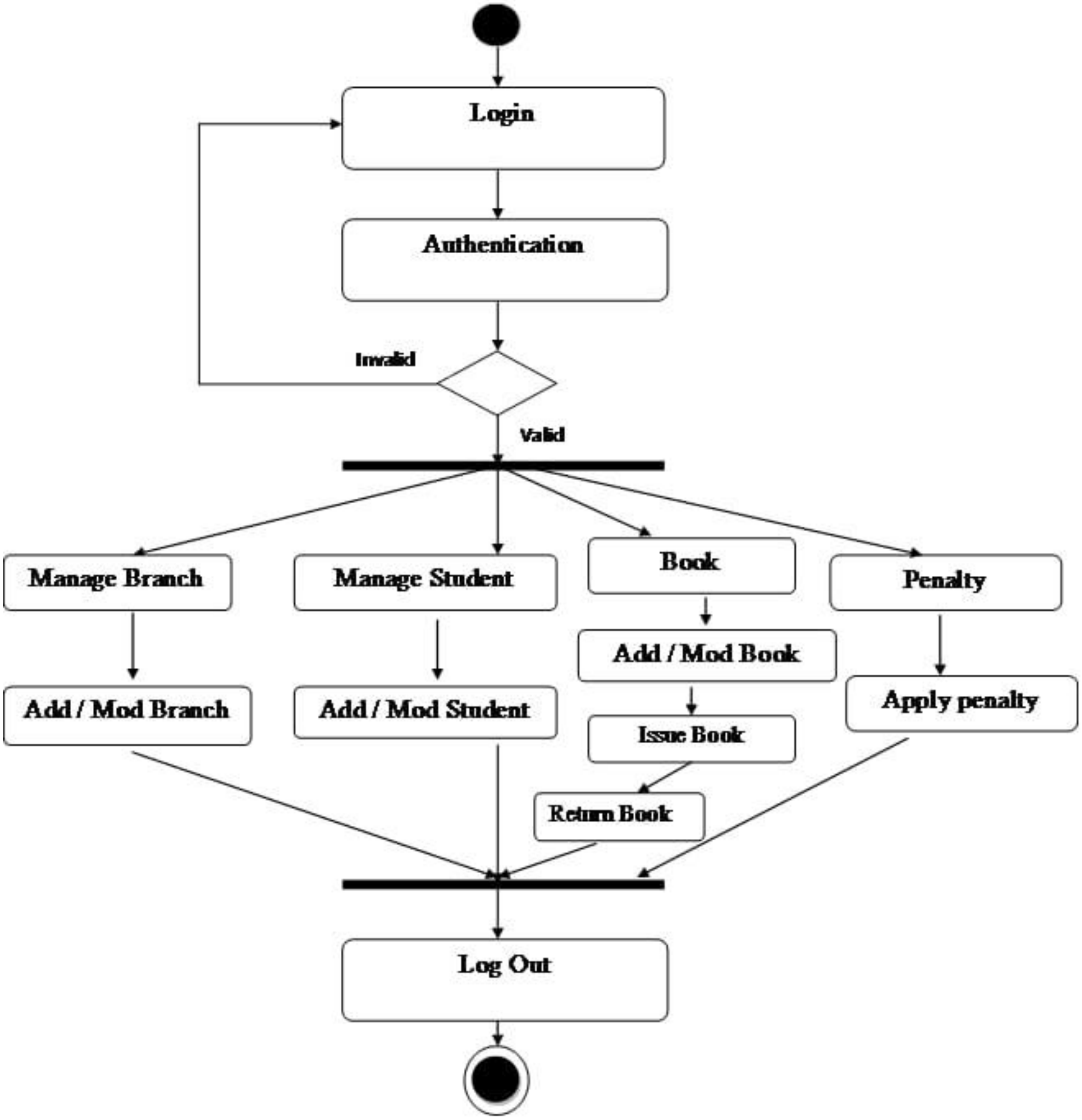
Use case diagram



Class Diagram




Member
Librarian
Book
Transaction
request a book
check availability
available
validate
check No of issued books
issue a book
<<create>>
add memeber and book details
update status
update member record
Member
Librarian
Book
Transaction


Login
Authentication
Invalid
Valid
Manage Branch
Manage Student
Book
Penalty
Add / Mod Book
Add / Mod Branch
Add / Mod Student
Apply penalty
Issue Book
Return Book
Log Out

Functions of a library:

1. Reusability: Libraries offer pre-implemented functions and modules that developers can leverage in their projects, saving time and effort in coding commonly needed functionalities.
2. Modularity: Libraries promote modular programming by breaking down complex functionalities into smaller, manageable units. This makes code easier to maintain, test, and debug.
3. Encapsulation: Libraries encapsulate their functionalities, hiding the internal implementation details from the user. This abstraction allows developers to focus on using the library's features without needing to understand how they are implemented.
4. Standardization: Libraries often adhere to established standards and best practices, ensuring consistency and reliability across different projects and environments.
5. Performance Optimization: Well-designed libraries are optimized for performance, helping developers achieve better efficiency and scalability in their applications.

Non-functions of a library:

6. Business Logic: Libraries typically do not implement application-specific business logic. They provide generic functionalities that can be applied across different projects but do not dictate how these functionalities should be used in specific business contexts.
7. User Interface: Libraries generally do not handle user interface (UI) components directly. While some libraries may offer UI

components or widgets, their primary focus is on providing backend functionality rather than designing user interfaces.

8. Application State Management: Libraries do not manage the overall state of an application. State management is typically handled by application frameworks or specific state management libraries, separate from the functionalities provided by general-purpose libraries.
9. Integration: While libraries facilitate integration with other software components, they do not manage the entire integration process. Developers are responsible for integrating libraries into their projects and ensuring compatibility with other components.
10. Business Model: Libraries themselves do not define or enforce any particular business model. While some libraries may be open source, others may have commercial licenses or restrictions. However, the business model is typically determined by the organization or individuals who develop and distribute the library rather than inherent to the library itself.