

Report for App-Dev 1 :

The system consists of several key models to represent the entities and relationships within the library ecosystem:

- **User Model:** Represents users of the system, including both general users and librarians. Each user has attributes such as username, password (hashed for security), and role (e.g., user or librarian).
- **Section Model:** Represents the different sections or categories in the library, such as fiction, non-fiction, reference, etc. Each section has attributes including ID, name, date created, and description.
- **Book Model:** Represents individual e-books available in the library. Each book has attributes such as ID, name, content, author(s), date issued, and return date. Additionally, it has relationships with the Section Model to associate books with specific sections.

The system follows a client-server architecture, with the backend implemented using the Flask framework and the frontend using Jinja2 templates with Bootstrap for styling. The system's components include:

- **Backend (Flask):** The Flask application serves as the backend logic for handling user requests, database interactions, and business logic. It consists of routes for different functionalities such as user authentication, CRUD operations on sections and books, and handling book requests, returns, and access management. The application interacts with the SQLite database to store and retrieve data.
- **Frontend (Jinja2 Templates + Bootstrap):** The frontend comprises HTML templates generated using Jinja2, combined with Bootstrap for styling and layout. Templates are designed to provide users with intuitive interfaces for logging in, browsing sections and books, making book requests, and managing user profiles. Forms are used for user input, with client-side validation to ensure data integrity.
- **Database (SQLite):** The SQLite database stores persistent data such as user information, section details, and book records. Tables are designed to maintain data consistency and enforce relationships between entities. The database schema is structured to support efficient querying and retrieval of information.

Conclusion:

The Library Management System employs a robust set of models and a well-defined system design to provide users with a seamless experience for accessing and managing e-books. By leveraging Flask for backend logic, Jinja2 templates with Bootstrap for frontend development, and SQLite for data storage, the system offers a scalable and efficient solution for library management needs. With user-friendly interfaces, intuitive navigation, and secure data handling, the system aims to streamline library operations and enhance user satisfaction.