Project Documentation: Library Management System

# Project Overview

The Library Management System is designed to manage and facilitate operations in a library, including user authentication, book searching, borrowing, and reservations. The system uses an API Gateway to handle requests from users and routes them to appropriate services for processing. It interacts with a database to perform CRUD operations and manage data.

### ****Key Components****

 **API Gateway**: The entry point for all API requests. It routes requests to the appropriate service and handles responses.

 **AuthService**: Handles user authentication, including validating credentials and issuing authentication tokens.

 **BookService**: Manages book-related operations, such as searching for books.

 **BorrowingService**: Manages book borrowing operations, including updating book statuses.

 **ReservationService**: Handles book reservation requests and updates reservation records.

 **Database**: Stores user data, book information, borrowing records, and reservations.

# System Flow

### ****1 User Authentication****

1. **User** sends login credentials (username and password) to the **API Gateway**.
2. **API Gateway** forwards the credentials to the **AuthService**.
3. **AuthService** queries the **Database** to validate the credentials.
4. **Database** returns user information to the **AuthService**.
5. **AuthService** issues an authentication token and sends it back to the **API Gateway**.
6. **API Gateway** returns the authentication token to the **User**.

### ****.2 Book Search****

1. **User** sends a search request with the book title to the **API Gateway**.
2. **API Gateway** forwards the request to the **BookService**.
3. **BookService** queries the **Database** for books matching the title.
4. **Database** returns book information to the **BookService**.
5. **BookService** sends the book information to the **API Gateway**.

### ****3 Borrow Book****

1. **User** requests to borrow a book by ID through the **API Gateway**.
2. **API Gateway** forwards the request to the **BorrowingService**.
3. **BorrowingService** updates the book status in the **Database** to "borrowed".
4. **Database** confirms the status update to the **BorrowingService**.
5. **BorrowingService** sends a success message to the **API Gateway**.
6. **API Gateway** confirms the borrowing success to the **User**

### ****4 Reserve Book****

1. **User** requests to reserve a book by ID through the **API Gateway**.
2. **API Gateway** forwards the request to the **ReservationService**.
3. **ReservationService** creates a reservation record in the **Database**.
4. **Database** confirms the reservation creation to the **ReservationService**.
5. **ReservationService** sends a success message to the **API Gateway**.
6. **API Gateway** confirms the reservation success to the **User**.

### ****5**** ERD (Entity-Relationship)

User: Represents library users/admins.

Attributes: id, name, email, password, role (admin/user), created\_at, updated\_at.

Book: Represents the books in the library.

Attributes: id, title, author, isbn, published\_at, created\_at, updated\_at.

Category: Represents book categories.

Attributes: id, name, created\_at, updated\_at.

BookCopy: Represents different copies of the same book.

Attributes: id, book\_id, copy\_number, status (available/borrowed), created\_at, updated\_at.

Borrowing: Tracks which user borrowed which book copy.Attributes: id, user\_id, book\_copy\_id, borrowed\_at, due\_date, returned\_at, created\_at, updated\_at.

Reservation: Tracks user reservations of books.

Attributes: id, user\_id, book\_id, reserved\_at, status (active/canceled), created\_at, updated\_at.

### ****6****) Sequence Diagram

