Library Management System – ER Diagram Documentation

This documentation describes the conceptual data model of an enhanced Library Management System (LMS). The model was designed after reviewing an early version of the client's Flask-based application and discussing potential improvements to meet the real-world operational needs of a library. The goal of this ER model is to support robust and flexible data handling for common library functionalities such as managing books, tracking borrowing and returns, handling reservations, and supporting user authentication with multiple roles. It emphasizes data normalization, scalability, and clear relationships between entities.

Objectives of the Redesign

After evaluating the original system, several key functional and data requirements were identified in consultation with the client:

- Enable tracking of borrowed and returned books, including due dates and return status.

- Allow books to have multiple authors and be assigned to multiple categories or genres.

- Implement a reservation system for books that are currently checked out.

- Separate system users (login credentials and roles) from member-specific details.

- Include support for different user roles: administrators, librarians, and members.

- Store contact information for library members.

These insights directly shaped the structure of the ER diagram.

Entity Overview and Relationships

1. User

The `User` entity represents all individuals who have access to the system. Each user has a unique login with a username and password hash and is assigned a specific role (`admin`, `librarian`, or `member`). This role determines their access privileges within the system.

2. Member

A `Member` represents an individual who borrows or reserves books from the library. Each member is associated with exactly one `User` account, forming a one-to-one relationship. Additional member-specific details such as full name, email, phone number, and address are stored separately from authentication credentials to keep the data well-structured and secure.

3. Book

The Book entity captures bibliographic and inventory data for all books in the library. This includes the title, ISBN, publisher, year of publication, and the number of total and available copies. A book can be linked to one or multiple authors and categories.

4. Author

The `Author` entity contains basic information about book authors. A many-to-many relationship is established between `Book` and `Author` via the `BookAuthor` junction table, allowing for proper modeling of books written by one or multiple contributors.

5. Category

Books are organized into one or more `Category` entries (e.g., Fiction, History, Science). The many-to-many relationship between books and categories is handled through the `BookCategory` table.

6. Borrow

The `Borrow` entity logs when a member checks out a book. It includes references to both the `Book` and `Member` involved in the transaction, along with the borrow date, due date, and return date. This enables tracking of loan activity and overdue items.

Relationship Summary

Relationship	Type
Book ↔ Author	Many-to-Many
Book ↔ Category	Many-to-Many
Member ↔ User	One-to-One
Member ↔ Borrow	0-to-Many
Book ↔ Borrow	0-to-Many