**DOCUMENTATION FOR LIBRARY MANAGEMENT SYSTEM**

The Library Management System is a project designed to showcase skills in object-oriented programming, data structures, database design, SQL, and JavaFX. This system provides efficient management of library resources and operations, offering a user-friendly interface for librarians and patrons to interact with the library's collection.

**System Overview**

The Library Management System will help librarians to manage books, patrons, and transactions like borrowing and returning books. Patrons can search for books, borrow them, and check their borrowing history. The system will feature a JavaFX-based graphical user interface, and a MySQL database will be used to store all the library data.

**Project Requirements and Objectives**

**Objectives**

* Implement a comprehensive library management system.
* Apply object-oriented programming principles for flexible and scalable code.
* Utilize essential data structures to optimize library operations.
* Design and implement a normalized database schema.
* Develop SQL skills for efficient database manipulation.

**IDE and tools**

* Java Development Kit (JDK)
* IntelliJ IDEA (IDE)
* MySQL (Database)
* JavaFX (User Interface)

**Functional Requirements**

* **User Management:**
  + Librarians can add, update, and delete patron records.
  + Patrons can update their own profiles.
* **Book Management:**
  + Librarians can add, update, and delete book records.
  + Patrons can search for books by title, author, genre, and availability.
* **Transaction Management:**
  + Librarians can record the borrowing and returning of books.
  + Patrons can view their borrowing history.
* **Reservation Management:**
  + Patrons can reserve books currently on loan.
  + Librarians can manage book reservations.
* **Reports:**
  + Librarians can generate reports on book availability, borrowed books, and patron activities.

**Non-Functional Requirements**

1. **Performance:** The system should be responsive and handle multiple transactions efficiently.
2. **Reliability:** The system should be robust with minimal downtime and data loss.
3. **Usability:** The user interface should be intuitive and user-friendly.
4. **Scalability:** The system should support the addition of new features and handle increasing amounts of data.
5. **Security:** The system should have secure authentication and authorization mechanisms to protect user data and library records.

**User Roles**

**Patron**

* Search for books in the library catalog.
* Borrow books if available.
* Return books within the borrowing period.
* Reserve books that are currently on loan.
* View borrowing history and current loans.
* Update profile information.

**Librarian**

* Add, update, and delete patron records.
* Add, update, and delete book records.
* Manage borrowing and returning of books.
* Manage book reservations.
* Generate reports on library operations.

**Use Cases**

**Use Case: Add Book**

**Actor:** Librarian  
**Description:** Librarian adds a new book to the library.  
**Steps:**

* Librarian selects Add Book from the menu.
* Librarian enters book details (title, author, number of copies).
* System validates the entered details.
* System saves the new book in the database.
* System confirms the addition of the new book.

**Use Case: Borrow Book**

**Actor:** Patron  
**Description:** Patron borrows an available book from the library.  
**Steps:**

* Patron searches for a book.
* Patron selects an available book to borrow.
* System records the borrowing transaction.
* System updates the book's status to "borrowed".
* System confirms the borrowing transaction.

**Use Case: Return Book**

**Actor:** Patron  
**Description:** Patron returns a borrowed book to the library.  
**Steps:**

* Patron selects the book to return from their borrowing history.
* System records the return transaction.
* System updates the book's status to "available".
* System checks if there are reservations for the book.
* System confirms the return transaction.

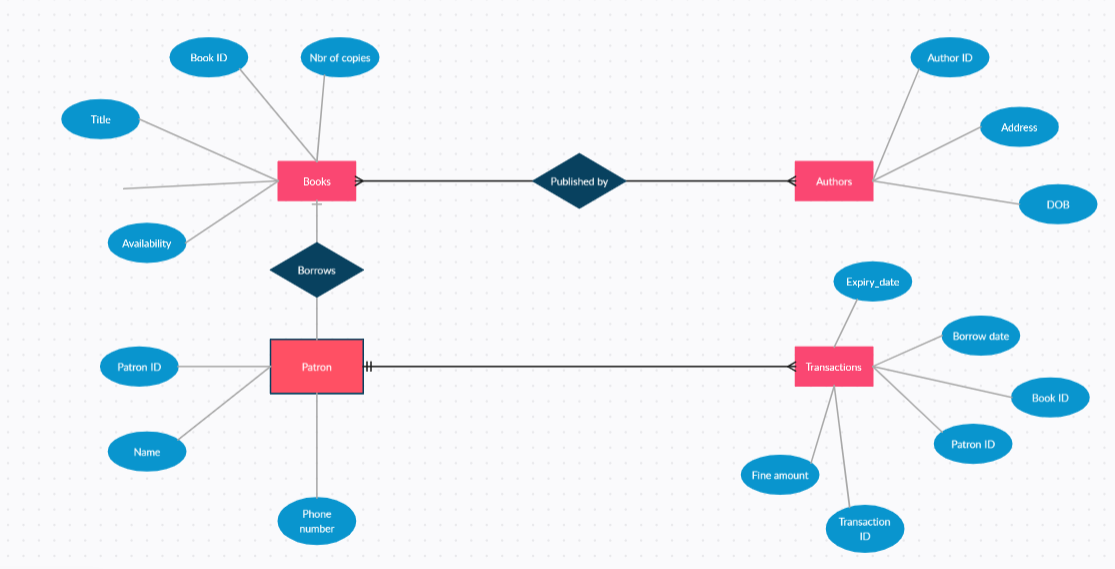
**Use Case: Reserve Book**

**Actor:** Patron  
**Description:** Patron reserves a book that is currently on loan.  
**Steps:**

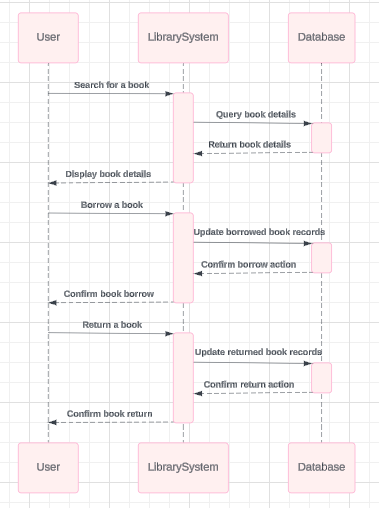
1. Patron searches for a book.
2. Patron selects a book to reserve.
3. System checks if the book is already reserved.
4. System records the reservation.
5. System confirms the reservation.

**Entity-Relationship Diagram (ERD)**

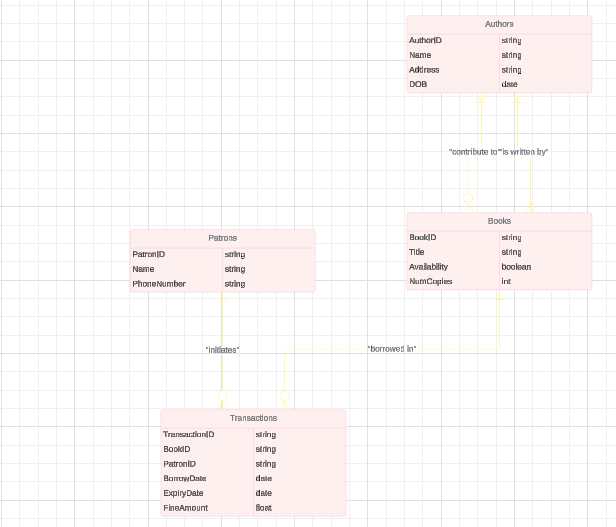
The Entity-Relationship Diagram (ERD) for the Library Management System is designed to show the functionalities of book borrowing, tracking, and management within a library. It features key entities such as Books, Authors, Patrons, and Transactions, each with its attributes. The ERD highlights a many-to-many relationship between Books and Authors and a one-to-many relationship between Patrons and Transactions, capturing the borrowing process. The Books entity includes details like BookID, Title, Availability, and Number of Copies, while Authors includes AuthorID, Name, Address, and DOB. Patrons are identified by PatronID, Name, and PhoneNumber. Transactions track the borrowing activities with attributes such as TransactionID, BookID, PatronID, BorrowDate, ExpiryDate, and FineAmount. This diagram ensures data integrity and facilitates efficient management and retrieval of library records.

****

**Sequence Diagram**

****

**UML Diagrams**

****

**Conclusion**

This system will allow librarians to manage books and patrons, handle borrowing and returning transactions, and make reservations. Patrons can search for and reserve books, borrow and return them, and view their borrowing history.