

Project Requirements Document (PRD) - Advanced Library Management System

1. Project Overview

The **Advanced Library Management System** aims to streamline and automate the operations of a library. The project focuses on developing a scalable, efficient, and user-friendly system, leveraging object-oriented programming, data structures, database design, and SQL skills. This system will use **JavaFX** for the user-friendly interface for interacting with the library management system, providing features such as forms for data input, tables for displaying information, and buttons for executing actions and **JDBC** to interact with a relational database.

2. Objectives

- Develop a comprehensive library management system covering essential library functions.
- Implement object-oriented programming principles to achieve modular, reusable, and scalable code.
- Use fundamental data structures for optimized library operations.
- Design and implement a relational database schema that meets third normal form (3NF) requirements.
- Gain experience in writing efficient SQL queries for data manipulation.
- Build familiarity with JavaFX for creating interactive user interfaces.

3. Scope of Work

- Core library operations: adding/updating books and patrons, borrowing, returning, and reserving books.
- User interface design using JavaFX for basic interactions.
- Database integration using MySQL or PostgreSQL with JDBC.
- Role-based access control for librarians and patrons.
- Implementation of basic reporting functions.

4. Functional Requirements

Book Management

- **Add Book:** Users can add a new book with relevant details like ISBN, title, author, and genre.
- **Update Book:** Users can modify existing book information.
- **Delete Book:** Users can delete a book record from the system.
- **View Books:** Users can view all available books with options to filter and search.

- **Check Book Availability:** Users can check the status of a book (Available, On Loan, Reserved).

Patron (User / Librarian) Management

- **Add Patron:** Patrons can sign-up as new members/add users with details like name, email, and membership type.
- **Update Patron:** Users can modify their profile information.
- **Delete Patron:** Users can delete their profiles/ Librarians delete user profiles.
- **View Patron Records:** Users can view patron details and their borrowing history.

Transaction Management

- **Borrow Book:** Users can initiate book loans.
- **Return Book:** Users can process book returns and update the availability status.
- **Renew Book:** Users can extend the due date for borrowed books based on library policy.
- **Track Overdue Items:** System tracks overdue books and calculates fines if applicable.

Reservation Management

- **Reserve Book:** Users can reserve books if currently unavailable.
- **View Reservations:** Users can view the queue of reservations for each book.
- **Cancel Reservation:** Users can cancel a reservation request.

Authentication and Authorization

- **User Login:** Secure login system with roles (e.g., Librarian, Patron).
- **Access Control:** Limit functionality based on user roles (librarians vs. patrons).

5. Non-Functional Requirements

- **Usability:** The JavaFX interface should be intuitive and easy to navigate for users with basic computer skills.
- **Reliability:** System should handle database connection errors gracefully and prevent data loss.
- **Security:** Data access and operations should be secure, especially regarding user passwords and patron contact details.

6. User Requirements

- **Librarians:**

- Full access to book and patron management functions.
- Ability to process transactions and manage reservations.
- Access to reports and fine management.
- **Patrons:**
 - Limited access to their own borrowing history and the ability to reserve books.

7. Technical Requirements

- **Programming Language:** Java (JDK 21) with JavaFX for the front-end interface.
- **Database:** PostgreSQL, with the database normalized to 3NF.
- **Data Access Layer:** JDBC for database connections and queries.
- **Development Environment:** IntelliJ IDEA .

8. Data Requirements

Key Entities and Attributes:

- **Books:** book_id, isbn, title, author, genre, publication_date, status, copies_available.
- **Patrons:** patron_id, name, email, phone, address, membership_type.
- **Transactions:** transaction_id, book_id, patron_id, borrow_date, due_date, return_date, status.
- **Reservations:** reservation_id, book_id, patron_id, reservation_date, queue_position, status.
- **Librarian:** user_id, email, username, password.

9. Assumptions

- All patrons will have unique email addresses and IDs.
- Only one copy of each book can be borrowed per patron at a time.
- Users will have basic knowledge of operating a library system.