**System Requirements Document for Library Management System**

**Introduction**

The Library Management System (LMS) is designed to manage the operations of a library, including the cataloging of books, categories, reservations, borrowing and authors. The system will support the recording of book details, user information, and borrowing of books. The system will be built using JavaFX for the user interface and MySQL for database management, with JDBC for communication between the application and the database.

**Functional Requirements**

**1. User Roles and Permissions**

The system will have different user roles with specific permissions:

* **Librarian**:
  + Full access to all functionalities, including user management, book cataloging, loan management, fine management, and generating reports.
  + Can add, edit, or remove authors, categories, reservations, books, and borrowing.
  + Can manage book checkouts and returns, update book availability status, and manage overdue fines.
* **Patron (User)**:
  + Can borrow books, reserve books, view authors and categories and view their borrowing history.
  + Cannot modify the book catalog or user management.

**2. Book Management**

* **Book Information**:
  + **Book ID**: Auto-generated unique identifier for each book.
  + **Title**: Text field to input the title of the book.
  + **Author**: Text field to input the author's name.
  + **Publisher**: Text field to input the publisher's name.
  + **Genre**: Dropdown to select the genre of the book (e.g., Fiction, Non-fiction, Science, etc.).
  + **Quantity Available**: Numeric field indicating how many copies of the book are available.
  + **Total Available**: Numeric field indicating how many copies of the book are there in total.
  + **Location**: Text field for the book's shelf or section location within the library.
* **Form Features**:
  + **Add Book**: Button to add a new book to the catalog.
  + **Edit Book**: Button to modify the details of an existing book.
  + **Delete Book**: Button to remove a book from the catalog.
  + **Search Book**: Search bar to find books by title, author, or ISBN.

**3. User Management**

* **User Registration**:
  + Users must register with their details to borrow books.
  + Required fields: Name, Email, Account Type (Patron) and User ID (Auto-generated).
* **User Profile**:
  + Each user will have a profile containing their personal information, current borrowed books, and borrowing history.
* **Form Features**:
  + **Add User**: Option for administrators to register new users.

**4. Borrowing and Returning Books**

* **Borrowing Process**:
  + Users can borrow books for a predefined loan period (e.g., 14 days).
  + The system will track the books borrowed by each user, the due date, and the status of the book (borrowed or available).**5. Search Functionality**
* **Search Criteria**:
  + Users can search the catalog of books using various criteria such as title, author, genre, and ISBN.
  + The search results will display a list of matching books along with their availability status.
* **Search Form Features**:
  + **Search Bar**: Text field to input search terms.
  + **View Book Details**: Option to view detailed information for each book in the catalog.

**6. Transaction History**

* **Borrowing History**:
  + Each user will have a history of their previous borrowings, including the date borrowed, due date, return date, and fines (if any).**Report Generation**:
  + Reports can be filtered by date range and can be exported as a PDF or Excel file.

**Non-Functional Requirements**

* **Performance**:
  + The system should support concurrent access by multiple users with minimal performance degradation.
  + Response times for book searches, transaction history, and report generation should be under 3 seconds.
* **Security**:
  + The system should implement role-based access control to ensure users only have access to their authorized features.
  + Passwords should be encrypted, and all sensitive data should be protected.
* **Scalability**:
  + The system should be scalable to handle an increasing number of books, users, and transactions.
* **User Interface**:
  + The UI should be user-friendly and responsive, built using JavaFX with modern design elements for ease of use.