Business Requirements Document (BRD) for Library Management System (LMS)

# 1. Introduction

The purpose of this document is to provide a high-level overview of the **Library Management System (LMS)**, a web-based application designed to streamline the day-to-day operations of a library and enhance user experience. The LMS will facilitate the management of books, members, and borrowing/returning processes while offering a seamless interface for users to search and borrow books. The system will also allow members to track their borrowing history, reserve books, and view fines and due dates.

# 2. Project Goals

The primary objectives of the Library Management System are:

1. **Development of a Fully Functional Web Application:**
   * The LMS will be developed using **Spring Boot**, **React**, and **MySQL** for its backend, frontend, and database respectively.
2. **Efficient Management of Library Resources:**
   * Enable librarians (admin users) to efficiently manage the library’s books, members, and lending processes.
3. **Enhanced User Experience for Members:**
   * Provide a user-friendly interface for library members to easily search for books, borrow books, track borrowing history, and manage reservations.
4. **Scalability and Robustness:**
   * The system must be scalable and robust enough to handle a moderate number of books and members, ensuring high availability and reliability.

# 3. Key Features

The system will incorporate the following features:

## 3.1 User Management

### 3.1.1 Tasks

* Implement a simple user registration form in React.
* Create a backend endpoint to save user data to the users table.
* Add basic validation (e.g., required fields) in both frontend and backend.

### 3.1.2 Deliverable

* A working user registration feature with basic validation.

### 3.1.3 Further Enhancement

* **User Registration and Authentication:**
  + Users (both librarians and members) will be able to register an account and log in to the system.
* **User Roles:**
  + **Librarian (Admin):** Full administrative rights to manage books, members, and borrowing transactions.
  + **Member:** Can borrow and return books, search for books, view borrowing history, and manage account details.
* **Profile Management:**
  + Users can update their profile details (e.g., name, contact information, etc.).
* **Password Reset:**
  + Users can reset their password via email if forgotten or if a change is required.

## 3.2 Member Management (Further Enhancement)

* **Member Registration:**
  + Librarians can register new members with essential personal details (name, address, contact information, etc.).
* **Member Information Management:**
  + Librarians can edit member information or delete members from the system as necessary.
* **Member Search:**
  + Librarians can search for members using parameters like name or ID to view details or perform actions.

## 3.3 Book Management

### 3.3.1 Tasks

* Implement CRUD operations for book management (add, update, delete, view).
* Add search and filter functionality for books (e.g., by title, author, category).
* Track book status (available, borrowed, reserved).

### 3.3.2 Deliverable

* A fully functional book management system for admins.

### 3.3.3 Further Enhancement

* **Book CRUD Operations:**
  + Librarians can add new books, update existing books, delete books, and view book details.
* **Book Details:**
  + Each book will have details such as **ISBN**, **title**, **author**, **category**, **publication year**, and **copies available**.
* **Search and Filter Books:**
  + Members and librarians can search and filter books by various criteria such as title, author, category, or publication year.
* **Track Book Status:**
  + The system will track the current status of a book (e.g., available, borrowed, reserved).

## 3.4 Lending Management

### 3.4.1 Tasks

* Implement book search with advanced filters (e.g., by category, publication year).
* Add borrowing and returning functionality with due date tracking.
* Display borrowing history for members.

### 3.4.2 Deliverable

* A fully functional book search, borrow, and return system.

### 3.4.3 Further Enhancement

* **Borrowing and Returning Books:**
  + Members can borrow and return books through the system.
* **Borrowing History:**
  + Members can view their past borrowing history, including borrowed books, return dates, and fines.
* **Reserve Books:**
  + Members can reserve books that are currently unavailable.
* **Due Date Tracking:**
  + The system will track due dates for borrowed books and notify users accordingly.
* **Fine Calculation:**
  + If books are overdue, the system will automatically calculate fines based on the defined rules.

# 4. Business Rules

The following business rules will govern the system’s operation:

## 4.1 Tasks

* Implement scheduled tasks to check for overdue books.
* Display overdue notifications in the UI.
* Calculate fines for overdue books.

## 4.2 Deliverable

* A fully functional overdue notification and fine calculation system.

## 4.3 Further Enhancement

### 4.3.1 Membership Rules

* **Borrowing Limit:**
  + A member can borrow a maximum of **3 books** at a time.
* **Membership Validity:**
  + A member’s membership is valid for **1 year** from the date of registration.
* **Active Membership Requirement:**
  + Only members with valid, active memberships can borrow books.

### 4.3.2 Lending Rules

* **Loan Duration:**
  + The standard loan period for a book is **14 days**.
* **Renewal Limit:**
  + A member can renew a borrowed book a maximum of **2 times**.
* **Overdue Books:**
  + A member cannot borrow additional books if they have any **overdue** books.
* **Fine Limit:**
  + A member cannot borrow books if their accumulated fines exceed **$10**.

### 4.3.3 Fine Calculation Rules

* **Overdue Fine:**
  + A fine of **$0.50 per day** will be applied for each day a borrowed book is overdue.
* **Fine Accumulation:**
  + The fine will start accumulating from the **day after the due date**.
* **Maximum Fine per Book:**
  + The maximum fine that can be accumulated per book is **$20**.

# 5. Integration, Testing & Final Documentation

## 5.1 Tasks:

Integrate frontend and backend APIs.

Perform unit and integration testing.

## 5.2 Deliverable

A fully integrated, tested project with complete documentation (BRD, SRS, API docs)

# 6. Technical Requirements

To meet the project goals and support the key features, the LMS will incorporate the following technical components:

* **Backend:**
  + Developed using **Spring Boot** to ensure scalability, security, and fast performance.
* **Frontend:**
  + Developed using **React** for a responsive and interactive user interface.
* **Database:**
  + **MySQL** will be used to store user information, book details, member borrowing history, and fine calculations.
* **Authentication and Authorization (Optional):**
  + User authentication will be handled using **JWT tokens** for secure login and role-based access control (RBAC).
* **Cloud Deployment (Optional):**
  + The application can be deployed on a cloud service like **AWS** for better scalability and availability.

# 7. Conclusion

This **Library Management System (LMS)** will significantly streamline the library’s operations and improve the user experience for both librarians and members. By providing a comprehensive solution for book management, member management, and lending operations, the system will create an efficient and scalable platform for modern libraries. With the integration of user-friendly features such as book search, borrowing history, fine calculation, and reservations, the LMS will provide a robust solution to meet the needs of both library staff and users.

This document outlines the high-level goals and requirements for the system, which will be further detailed during the design and development phases.