

# Business Requirements Document (BRD)

## 1. Project Overview

The **Library Management System (LMS)** is a web-based application aimed at streamlining library operations by providing automated management of books, users, and borrowing processes. The system ensures efficient book tracking, enhances the user experience, and reduces errors associated with manual library management.

## 2. Business Objectives

- Automate book tracking, borrowing, and returning processes.
- Provide a user-friendly interface for both librarians and members.
- Ensure secure and role-based access to the system.
- Enable real-time notifications for overdue books and fine calculations.
- Maintain comprehensive borrowing history and advanced search capabilities.
- Ensure seamless integration of front-end and back-end components.

## 3. Scope of Work

The **Library Management System** will include the following core functionalities:

### User Management

- User registration and login.
- Role-based access control (Librarian vs. Member).
- Password reset functionality.

### Book Management (Admin Role)

- CRUD operations for book management (Add, Update, Delete, View).
- Search and filter books by title, author, and category.
- Track book status (Available, Borrowed, Reserved).

### Book Search & Borrowing

- Advanced search with filters (Category, Publication Year, Availability).
- Borrowing and returning functionality with due date tracking.
- Display borrowing history for members.

### Overdue Notifications & Fine Calculation

- Scheduled tasks to check overdue books.
- Display overdue notifications in the UI.
- Automatic fine calculation based on overdue duration.

## **Security & Authentication**

- JWT-based authentication.
- Secure password storage and validation.
- Role-based access control.

# **4. Technical Requirements**

## **Backend**

- Spring Boot framework.
- RESTful API architecture.
- MySQL database.
- JPA/Hibernate for database operations.
- Spring Security for authentication.

## **Frontend**

- React.js with React Router for navigation.
- React Hooks for state management.
- Bootstrap or Material-UI for styling.
- Axios for API communication.

## **Database Requirements**

- Proper relationship mapping (Users, Books, Borrowings).
- Referential integrity enforcement.
- Indexing for optimized queries.

## **Testing & Documentation**

- Unit and integration testing.
- Comprehensive documentation including BRD, SRS, API documentation.

# **5. Business Rules**

## **Membership Rules**

- Members can borrow a maximum of 3 books at a time.
- Membership is valid for 1 year from the date of registration.
- Members must have a valid membership to borrow books.

## Lending Rules

- Loan duration: 14 days.
- Maximum of 2 renewals per book.
- Borrowing restrictions for overdue books or fines exceeding \$10.

## Fine Calculation

- \$0.50 per day for overdue books.
- Fine accrual starts from the day after the due date.
- Maximum fine per book: \$20.

## 6. Project Milestones

Milestone	Tasks	Duration
Project Setup	Set up Spring Boot, React, MySQL with navigation	4 hrs
User Registration & Authentication	Implement login, role-based access, password reset	6 hrs
Book Management (Admin)	CRUD operations, search & filter books, track status	8 hrs
Book Search & Borrow/Return	Implement advanced search, borrowing/returning, history	10 hrs
Overdue Notifications	Implement notifications and fine calculation	4 hrs
Integration & Testing	Integrate front-end & back-end, unit testing, documentation	8 hrs

## 7. Deliverables

- Fully functional Library Management System.
- Source code (React frontend and Spring Boot backend).
- Database scripts (ERDs, SQL schema, seed data).
- Documentation (BRD, SRS, API docs, user manual).
- Presentation materials.

## 8. Acceptance Criteria

- All core features must be functional and tested.
- The system should follow best coding practices and security standards.
- Documentation should be complete and up to date.
- The system should handle all role-based access control correctly.
- Fine calculation and overdue notifications should work as expected.

## 9. Conclusion

The **Library Management System** will provide an efficient and modernized solution to traditional library operations, enhancing both the librarian and member experience through automation and intuitive UI design. By integrating React.js, Spring Boot, and MySQL, the system will offer a scalable and robust platform to handle key library functionalities effectively.