**Business Requirements Document (BRD)**

1. **Overview**

The Library Management System is designed to provide an easy-to-use platform for library operations. The system will enable users to search for books, borrow and return them, and receive overdue notifications. Administrators will have the ability to manage the library inventory, including adding, updating, and deleting books. This system will be implemented using React for the frontend, Spring Boot for the backend, and MySQL for data storage.

1. **Objectives**

* Enable users to register, log in, and search for books.
* Allow users to borrow and return books with due date tracking.
* Implement role-based access for members and administrators. (**intermediate**)
* Provide administrative functions for managing books (create, update, delete).
* Alert users regarding overdue books and calculate fines for overdue returns (**intermediate**).
* Ensure smooth integration of the frontend and backend with comprehensive documentation.

1. **Stakeholders**

* **End Users**: Library members who want to search, borrow, and return books.
* **Administrators**: Library staff managing the inventory and book records.
* **Developers**: Learners building and maintaining the system as part of their educational journey.

1. **Scope**

**In Scope:**

* User authentication (registration, login, password reset (**intermediate**), role-based access control) (**intermediate**).
* Book search with filtering options (title, author, category, publication year) (**intermediate**).
* Book borrowing and returning with status tracking (available, borrowed, reserved).
* Administrative functionalities for managing book inventory (CRUD operations (**intermediate**)).
* Overdue notifications and fine calculations for late returns (**intermediate**).
* Comprehensive integration of frontend and backend APIs.
* Basic testing and documentation (Beginner Tier) with comprehensive unit and integration testing (**Intermediate**).

**Out of Scope:**

* Multi-branch management for libraries.
* Integration with third-party APIs for external book sources.
* Advanced analytics and reporting.

1. **Assumptions & Constraints**

* The project is for educational purposes and not intended for production use.
* Implementation is limited to MySQL, Spring Boot, and React.
* Data security measures are simplified for learning purposes.
* Basic due date tracking will be implemented, but real-time notifications may not be highly optimized.
* Scheduled tasks for checking overdue books will be implemented in the intermediate tier.

1. **Requirements & Features**
2. **User Registration & Authentication**
   * User registration and login
   * Role-based access control (Librarian vs. Member)
   * Backend endpoint to store user data
   * Basic validation on frontend and backend
   * Password reset functionality (**intermediate**)
3. **Book Management (Admin)**
   * Full CRUD (add, update, delete, view) operations for books (**intermediate**)
   * Form to add book details
   * Backend API to store book data
   * Display book list in UI
   * Advanced search and filters (title, author, category) (**intermediate**)
   * Track book status (available, borrowed, reserved)
4. **Book Search & Borrow/Return**
   * Search bar to search books by title
   * Search with advanced filters (category, publication year) (**intermediate**)
   * Borrow button to update book status
   * Borrowing and returning with due date tracking
   * Display borrowing history for users (**intermediate**)
5. **Overdue Notifications (Optional)**
   * Scheduled tasks to check for overdue books (**intermediate**)
   * Display overdue notifications
   * Calculate fines for overdue books (**intermediate**)
6. **Integration, Testing & Documentation**
   * Connect frontend and backend
   * Basic manual testing
   * Integrate frontend and backend APIs
   * Perform unit and integration testing
   * Write BRD, SRS, API documentation