# Business Requirements Document (BRD)

Document Version: 1.0  
Date: \_\_\_\_\_\_\_\_\_\_\_\_  
Prepared by: \_\_\_\_\_\_\_\_\_\_\_\_

## 1. Executive Summary / Business Objective

The Library Management System is a full-stack web application designed to digitize and automate core library operations. It aims to help librarians efficiently manage books, members, and lending processes, while also providing members with features to search, reserve, and track their borrowing history.

## 2. Stakeholders

- Project Sponsor  
- Librarians  
- Library Members  
- Development Team  
- QA/Testers

## 3. Scope of the System

In Scope:  
- User management with role-based access  
- Book catalog management  
- Lending and return tracking  
- Fine calculation and rule enforcement  
- Reservation system

Out of Scope:  
- Integration with third-party book APIs  
- Mobile app version

## 4. Business Requirements

- Allow user registration and role-based login.  
- Enable librarians to manage books and members.  
- Allow members to borrow, return, and reserve books.  
- Automatically calculate fines.  
- Enforce lending policies and restrictions.

## 5. Assumptions & Constraints

- Users must have internet access.  
- The system will run on modern web browsers.  
- Backend and database will be hosted on a secure server.

## 6. Success Metrics

- System handles 100 concurrent users.  
- 95% of test cases pass with no critical bugs.  
- Librarian operations speed improved by 50%.  
- 90% user satisfaction rate in feedback survey.

# Software Requirements Specification (SRS)

Document Version: 1.0  
Date: \_\_\_\_\_\_\_\_\_\_\_\_  
Prepared by: \_\_\_\_\_\_\_\_\_\_\_\_

## 1. Introduction

This document describes the functional and non-functional requirements for the Library Management System. It is intended for developers, testers, and stakeholders.

## 2. System Overview

The system is a web application composed of a React frontend, Spring Boot backend, and MySQL database. Authentication is managed through JWT tokens and role-based access control.

## 3. Functional Requirements

- User Registration/Login (JWT-based)  
- Role-based access: Member, Librarian  
- Book Management: Add/Edit/Delete/View/Search books  
- Member Management: Add/Edit/Delete/Search members  
- Lending: Borrow, Return, Renew books  
- Reservation: Reserve available books  
- Fine Calculation: $0.50/day overdue, capped at $20  
- Restrictions: Max 3 books, fines > $10 block borrowing

## 4. Non-Functional Requirements

- Performance: Pages load within 2 seconds  
- Security: Use HTTPS, JWT, hashed passwords  
- Scalability: Handle up to 500 members  
- Usability: Intuitive UI for both members and librarians

## 5. User Interfaces

- Login Page: Email/password fields  
- Dashboard: Role-based views  
- Book List: Search, filter, and actions  
- Member Details: Personal info and borrowing history  
- Lending Panel: Borrow, return, fine summary

## 6. System Requirements

- Frontend: React, Bootstrap  
- Backend: Spring Boot, REST API  
- Database: MySQL  
- Auth: JWT, Spring Security  
- Hosting: Cloud server with database access

## 7. Assumptions and Dependencies

- Backend requires Java runtime environment.  
- MySQL must be installed and accessible.  
- System depends on internet access and modern browsers.