### **Software Requirements Specification (SRS)**

#### **Library Management System (LMS) - Beginner Tier**

#### **1. Introduction**

* **Purpose**: This document details the software requirements for the beginner-tier Library Management System.
* **Intended Audience**: Developers, testers, and project stakeholders.
* **Technologies Used**:
  + Backend: **Spring Boot, REST API**
  + Frontend: **React.js**
  + Database: **MySQL**

#### **2. Functional Requirements**

| **ID** | **Requirement** |
| --- | --- |
| FR-01 | The system must allow users to register with email and password. |
| FR-02 | The system must validate required fields during registration. |
| FR-03 | Admin must be able to add new books via a form. |
| FR-04 | Users must be able to search books by title. |
| FR-05 | Users must be able to borrow books and update the book status in the database. |
| FR-06 | The system should provide a basic UI for searching and borrowing books. |
| FR-07 | (Optional) The system should show an overdue notification for borrowed books based on a static due date. |

#### **3. Non-Functional Requirements**

| **ID** | **Requirement** |
| --- | --- |
| NFR-01 | The system should follow basic web security standards. |
| NFR-02 | The system should be lightweight and performant for a small dataset. |
| NFR-03 | The UI should be simple and intuitive. |
| NFR-04 | The backend should follow RESTful API principles. |
| NFR-05 | The database should enforce proper indexing for search efficiency. |

#### **4. System Architecture**

* **Frontend:** React.js (UI components, API integration)
* **Backend:** Spring Boot (RESTful API)
* **Database:** MySQL (Users & Books tables)

#### **5. Data Requirements**

| **Table** | **Fields** |
| --- | --- |
| **Users** | user\_id (PK), name, email, password |
| **Books** | book\_id (PK), title, author, status (available/borrowed) |

#### **6. API Endpoints**

| **Endpoint** | **Method** | **Description** |
| --- | --- | --- |
| /api/register | POST | Registers a new user |
| /api/books | GET | Retrieves all books |
| /api/books/search?title=xyz | GET | Searches books by title |
| /api/books/add | POST | Admin adds a new book |
| /api/books/borrow/{id} | PUT | Updates book status to borrowed |

#### **7. Security Requirements**

* Basic input validation (e.g., required fields).
* No advanced authentication in beginner tier.
* No sensitive data stored unencrypted.

#### **8. Assumptions & Limitations**

* User roles are minimal (Admin, Member).
* No support for multi-library management.
* UI design is kept basic.

#### **9. Testing Requirements**

| **Test Case** | **Description** |
| --- | --- |
| TC-01 | Ensure user registration works with valid inputs. |
| TC-02 | Verify that books can be added successfully. |
| TC-03 | Check if book search works with valid inputs. |
| TC-04 | Test borrowing functionality and status update. |