### **Business Requirements Document (BRD)**

#### **Library Management System (LMS)**

#### **1. Introduction**

##### **1.1 Purpose**

The purpose of this document is to define the business requirements for the **Library Management System (LMS)**. This system will provide a digital solution to manage library operations, including book cataloging, user registration, book lending, and overdue tracking.

##### **1.2 Scope**

The LMS will be a **full-stack web application** designed for **librarians and library members**. It will allow:

* **Librarians** to manage books, track lending records, and administer library members.
* **Library Members** to search for books, borrow, return, and track due dates.

##### **1.3 Business Objectives**

* Improve **efficiency** in managing books and members.
* Reduce manual paperwork and **human errors** in book tracking.
* Provide **real-time updates** on book availability.
* Automate overdue tracking and notifications.
* Enhance **user experience** with a user-friendly web interface.

##### **1.4 Assumptions & Constraints**

* The system will be **web-based** and accessible via browsers.
* Users must have **valid login credentials** to access functionalities.
* Only **librarians** can manage book inventory and users.
* System performance must support at least **100 concurrent users**.

#### **2. Business Requirements**

##### **2.1 Stakeholders**

| **Role** | **Responsibility** |
| --- | --- |
| **Library Administrator** | Manages books, members, and lending policies. |
| **Library Members** | Searches, borrows, reserves, and returns books. |
| **System Administrator** | Manages system settings and user roles. |

##### **2.2 Functional Requirements**

| **ID** | **Requirement Description** | **Priority** |
| --- | --- | --- |
| **FR1** | Users can register, log in, and reset passwords. | High |
| **FR2** | Librarians can **add, edit, delete, and search** books. | High |
| **FR3** | Members can **search and filter books** based on title, author, category. | High |
| **FR4** | Members can borrow books and track due dates. | High |
| **FR5** | Members can **renew or return books** before the due date. | Medium |
| **FR6** | System automatically **calculates overdue fines** and restricts borrowing if unpaid. | High |
| **FR7** | Librarians can track **borrowed and overdue books**. | High |
| **FR8** | Members receive **email notifications** for due dates and overdue books. | Medium |
| **FR9** | The system must support **role-based access control** (Librarians vs Members). | High |

##### **2.3 Non-Functional Requirements**

| **ID** | **Requirement Description** | **Priority** |
| --- | --- | --- |
| **NFR1** | System must be **accessible via web browsers** (Chrome, Firefox, Edge). | High |
| **NFR2** | System must support **100 concurrent users**. | Medium |
| **NFR3** | Login authentication must be secured with **JWT and bcrypt hashing**. | High |
| **NFR4** | The database must support **backup and recovery mechanisms**. | Medium |
| **NFR5** | The system should follow **WCAG 2.1 accessibility guidelines**. | Low |

#### **3. Business Process Workflow**

**3.1 User Registration & Authentication**

1. User registers with **name, email, and password**.
2. System verifies email and **activates the account**.
3. User logs in and is **assigned a role** (Librarian or Member).

**3.2 Book Borrowing Process**

1. Member searches for a book.
2. If available, they can **borrow it**.
3. The system records the **due date**.
4. Members can **return or renew** before the due date.

**3.3 Overdue Management**

1. System checks overdue books **daily**.
2. Overdue fines are **automatically calculated**.
3. Members receive **reminders via email**.

#### **4. Use Case Diagram**

**Actors:**

* **Librarian**
* **Member**
* **System**

**Use Cases:**

* Login/Logout
* Manage Books (CRUD)
* Search Books
* Borrow & Return Books
* Manage Members
* Overdue Tracking

📌 **(Attach UML Use Case Diagram here)**

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

| **Table Name** | **Attributes** |
| --- | --- |
| **Users** | id, name, email, password, role |
| **Books** | id, title, author, category, copies\_available |
| **Lending** | id, book\_id, member\_id, borrow\_date, return\_date, status |
| **Fines** | id, member\_id, amount\_due, paid\_status |

#### **6. Risk Assessment**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| **Data Loss** | High | Implement **database backups**. |
| **Security Breach** | High | Use **JWT authentication & password encryption**. |
| **System Downtime** | Medium | Deploy on **cloud-based infrastructure**. |

#### **7. Acceptance Criteria**

* All **core functionalities** (User registration, Book Management, Borrowing, Overdue) must work.
* System must be **secure and responsive** across devices.
* Automated **email notifications** for overdue books must function properly.
* System should be **tested with at least 50 users** before deployment.

#### **8. Project Timeline**

| **Milestone** | **Deliverables** | **Timeframe** |
| --- | --- | --- |
| **Project Setup** | Repo setup, Backend/Frontend init | **Week 1** |
| **User Authentication** | JWT-based login system | **Week 2** |
| **Book Management** | CRUD operations for books | **Week 3** |
| **Borrow & Return Features** | Lending system implementation | **Week 4** |
| **Overdue & Notifications** | Automated fine calculation & emails | **Week 5** |
| **Testing & Deployment** | Final QA and cloud hosting | **Week 6** |

### **9. Conclusion**

The **Library Management System (LMS)** will streamline **library operations** by automating book lending, tracking overdue books, and managing users. It will be built as a **secure, scalable, and user-friendly web platform**.