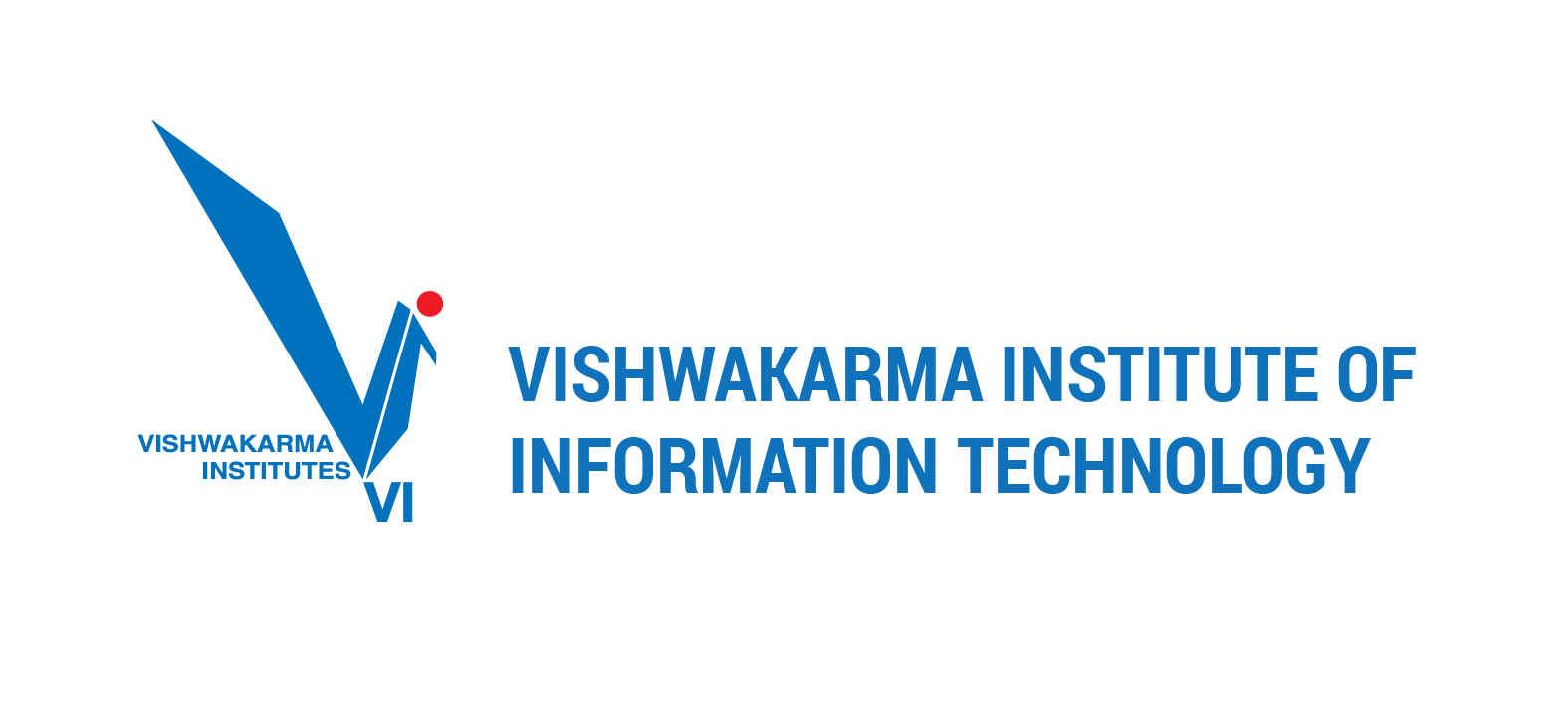
** Software Requirements**

**Specification**

**for**

**<Library Management System>**

**Version 1.0 approved**

**Prepared by <Pragati Pise, Surbhi Bagad, Unnati Lunawat>**

**<30/1/2025>**

**Table of Contents**

[**Table of Contents** **ii**](#_TOC_250032)

[**Revision History** **ii**](#_TOC_250031)

1. [**Introduction** **1**](#_TOC_250030)
   1. [Purpose 1](#_TOC_250029)
   2. [Document Conventions 1](#_TOC_250028)
   3. [Intended Audience and Reading Suggestions 1](#_TOC_250027)
   4. [Project Scope 1](#_TOC_250026)
   5. [References 1](#_TOC_250025)
2. [**Overall Description** **2**](#_TOC_250024)
   1. [Product Perspective 2](#_TOC_250023)
   2. [Product Features 2](#_TOC_250022)
   3. [User Classes and Characteristics 2](#_TOC_250021)
   4. [Operating Environment 2](#_TOC_250020)
   5. [Design and Implementation Constraints 2](#_TOC_250019)
   6. [User Documentation 2](#_TOC_250018)
   7. [Assumptions and Dependencies 3](#_TOC_250017)
3. [**System Features** **3**](#_TOC_250016)
   1. [System Feature 1 3](#_TOC_250015)
   2. [System Feature 2 (and so on) 4](#_TOC_250014)
4. [**External Interface Requirements** **4**](#_TOC_250013)
   1. [User Interfaces 4](#_TOC_250012)
   2. [Hardware Interfaces 4](#_TOC_250011)
   3. [Software Interfaces 4](#_TOC_250010)
   4. [Communications Interfaces 4](#_TOC_250009)
5. [**Other Nonfunctional Requirements** **5**](#_TOC_250008)
   1. [Performance Requirements 5](#_TOC_250007)
   2. [Safety Requirements 5](#_TOC_250006)
   3. [Security Requirements 5](#_TOC_250005)
   4. [Software Quality Attributes 5](#_TOC_250004)
6. [**Other Requirements** **5**](#_TOC_250003)

[**Appendix A: Glossary** **5**](#_TOC_250002)

[**Appendix B: Analysis Models** **6**](#_TOC_250001)

[**Appendix C: Issues List** **6**](#_TOC_250000)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

The **Library Management System (LMS)** is designed to automate and streamline library operations, replacing manual processes with a digital platform. The system will facilitate book cataloging, user management, borrowing and returning of books, and fine calculation for overdue books.

This SRS document defines the features and requirements of **LMS Version 1.0**. The system will be developed as a **web-based application** and can also be extended for desktop or mobile platforms. It will support different user roles, including **librarians, library members, and administrators**.

This document is intended for **developers, testers, library staff, project managers, and stakeholders** involved in the system's design, development, and implementation.

**1.2 Document Conventions**

This document follows **IEEE SRS 830-1998** standards and includes:

* **Formatting Conventions:**
  + **Bold**: Section headings and key terms.
  + *Italics*: Used for examples and references.
  + Monospace: Used for code snippets or commands.
* **Requirement Priorities:**
  + **Must-Have (M)** – Essential for system functionality.
  + **Should-Have (S)** – Important but not mandatory.
  + **Nice-to-Have (N)** – Features that can be included in future releases.
* **Diagrams and Tables:**
  + UML diagrams, flowcharts, class diagrams, state diagrams, ER diagrams, Kano diagrams, Use case diagrams, and tables will be used where necessary to illustrate system workflows.

**1.3 Intended Audience and Reading Suggestions**

This document is intended for the following groups:

* **Library Staff & Administrators**: To understand how the system will improve operations.
* **Developers**: To design, develop, and integrate the system features.
* **Testers**: To verify the system's functionality and compliance with requirements.
* **Project Managers**: To track progress and ensure the system meets business goals.
* **Stakeholders & End Users**: To review system capabilities and usability.

**Reading Suggestions**

* **New Readers & Stakeholders**: Start with **Section 1 (Introduction)** and **Section 2 (Overall Description)**.
* **Developers & Testers**: Focus on **Section 3 (Functional Requirements)** and **Section 4 (System Design & Architecture)**.
* **Library Staff**: Refer to **Section 5 (User Roles & Permissions)** and **Section 6 (Reports & Analytics)**.

**1.4 Project Scope**

**Overview**

The **Library Management System (LMS)** is a **centralized digital platform** that enables library staff to manage books and users efficiently. It helps members find and borrow books easily while maintaining accurate records of all transactions.

**Key Features & Functionality**

1. **Book Management**
   * Add, update, delete, and categorize books.
   * Maintain ISBN, author, and publisher details.
   * Track book availability and copies.
2. **User Management**
   * Register members and manage user accounts.
   * Assign roles (Librarian, Admin, Member).
   * Issue library cards and maintain user profiles.
3. **Borrowing & Returning**
   * Issue books to members.
   * Track due dates and overdue books.
   * Apply fines for late returns.
4. **Search & Reports**
   * Search books by title, author, category, or ISBN.
   * Generate reports on book usage, overdue books, and user activity.
5. **Multi-User Access**
   * Role-based permissions for Admins, Librarians, and Members.
6. **Security & Access Control**
   * Login authentication and password protection.
   * Data encryption and backup.
7. **Additional Features** (Future Enhancements)
   * **E-Book Integration**: Allow members to read digital books.
   * **Mobile App Support**: Enable mobile access.
   * **RFID-Based Book Issuance**: Automate borrowing using RFID tags.

**Business Goals & Benefits**

* Reduce manual work and improve efficiency.
* Minimize errors in book transactions.
* Enhance user experience through digital access.
* Improve library resource tracking and reporting.

**1.5 References**

Below are the key references and resources used for defining this system:

1. **IEEE Std 830-1998** – Software Requirements Specification Standard.
2. **Koha Library System** – Open-source LMS Documentation (<https://koha-community.org>).
3. **Evergreen Library System** – Documentation for Open-Source LMS (<https://evergreen-ils.org>).
4. **Library of Congress MARC Standards** – Cataloging and metadata standards (https://www.loc.gov/marc/).
5. **ISO 27001 Security Standards** – Guidelines for secure digital library systems.
6. **University Library Guidelines** – Example: Harvard Library System (<https://library.harvard.edu>).

**2. Overall Description**

**2.1 Product Perspective**

The **Library Management System (LMS)** is a **self-contained digital platform** designed to automate and streamline library operations, replacing traditional manual processes.

**Context & Origin**

* It is a **new system** that digitizes **book cataloging, member management, borrowing, and return tracking**.
* The LMS can function as a **standalone product** or be integrated into an **existing institutional IT system** (e.g., university or corporate library).

**System Integration & Interfaces**

* **Database**: Stores book details, member records, and transaction history.
* **User Interface**: A web-based or desktop application for interacting with the system.
* **External Systems** (Optional): Integration with **barcode scanners, RFID-based book tracking, and payment gateways** for fine collection.

**System Diagram**

A high-level system architecture may look like this:

📚 **Library Management System**

* 📁 **Database** (Book Records, User Data, Transactions)
* 🖥 **Librarian Interface** (Web/Desktop Application)
* 📲 **Member Interface** (Web Portal/Mobile App)
* 🔗 **External Systems** (Barcode/RFID, Payment Gateway, Third-Party Catalogs)

**2.2 Product Features**

The LMS offers the following **core features**:

✅ **Book Management**

* Add, update, delete books.
* Categorize books by genre, author, publisher, ISBN.
* Maintain book availability status.

✅ **User Management**

* Register and manage users (Members, Librarians, Admins).
* Assign roles and permissions.

✅ **Borrowing & Returning Books**

* Issue books to members with due dates.
* Track borrowed books and return deadlines.
* Calculate and apply fines for overdue books.

✅ **Search & Reporting**

* Search books by title, author, ISBN, or category.
* Generate reports (issued books, overdue books, library activity).

✅ **Security & Access Control**

* User authentication (login/logout).
* Data encryption and backup.

✅ **Optional Future Features**

* **E-Book Access** (for digital libraries).
* **RFID-based tracking** (automated check-in/check-out).
* **Mobile App Support** (for remote access).

**2.3 User Classes and Characteristics**

The LMS is designed for different user roles, each with varying levels of access and functionalities:

**👩‍🏫 Librarians & Staff**

* Manage books, track loans, and generate reports.
* Moderate user activity and issue fines.

**📖 Library Members (Students, Readers, Employees, etc.)**

* Search and borrow books.
* Check due dates and pay fines.

**🛠 System Administrators**

* Configure system settings.
* Manage user permissions and security.

**📊 Stakeholders (Library Managers, Decision-Makers)**

* Review system usage statistics.
* Analyze trends for future improvements.

**2.4 Operating Environment**

The LMS will operate in the following environment:

* **Hardware:**
  + Server: Cloud-based or on-premise.
  + Workstations: Standard desktop or laptop (Windows/Linux/macOS).
  + Mobile Devices: Android/iOS (for member access).
* **Software:**
  + **Operating System**: Windows, Linux (for servers), Web-based UI (any OS).
  + **Database**: MySQL, PostgreSQL, or MongoDB.
  + **Frontend Technologies**: HTML, CSS, JavaScript (React, Angular, or Vue.js).
  + **Backend Technologies**: Python (Django), Java (Spring Boot), Node.js.
* **Network Requirements:**
  + Internet/Wi-Fi connectivity for cloud-based systems.
  + Local network for offline operations (in a university or corporate setup).

**2.5 Design and Implementation Constraints**

The system will adhere to the following constraints:

**🚀 Technical Constraints**

* Database should support **high-volume transactions**.
* Must be **scalable** for future growth (supporting multiple libraries if needed).
* Must provide **secure user authentication (OAuth, JWT, or LDAP integration)**.

**📜 Regulatory & Compliance Constraints**

* Adherence to **library data management regulations**.
* Compliance with **GDPR (General Data Protection Regulation)** for user data.
* Follow **ISO 27001 security standards** for digital libraries.

**💻 System Integration Constraints**

* Should support integration with **barcode scanners and RFID**.
* Compatibility with **existing library systems** (if replacing an old system).

**2.6 User Documentation**

The following documentation will be provided:

📖 **User Manuals**

* **Librarian Guide** (Managing books, issuing returns, reports).
* **Member Guide** (Borrowing books, searching catalog, checking due dates).

🎓 **Online Help & Tutorials**

* Step-by-step video guides for new users.
* FAQs and troubleshooting documentation.

🛠 **Technical Documentation**

* API documentation for developers.
* Database schema and system architecture.

**2.7 Assumptions and Dependencies**

**📝 Assumptions**

* Users will have **basic computer literacy**.
* Library policies (borrowing limits, fine structure) will be predefined.
* The system will be accessed via a **stable internet connection** (for cloud-based deployment).

**🔗 Dependencies**

* The LMS may depend on:
  + **Third-party payment gateways** (for online fine payments).
  + **Existing digital library databases** (e.g., Koha, MARC cataloging).
  + **Security frameworks** (for user authentication and access control).

**3. System Features**

**3.1 User Authentication & Login**

**Description and Priority:**

* Users (students, teachers, and librarians) must log in before accessing the system.
* Priority: **High**

**Stimulus/Response Sequences:**

1. User enters their credentials (username and password).
2. System verifies credentials and grants/denies access.
3. If credentials are incorrect, the system displays an error message.

**Functional Requirements:**

* The system shall allow users to **register, log in, and log out** securely.
* The system shall support **role-based access control** (Admin, Librarian, Member).
* The system shall provide a **password reset** option.

**3.2 Catalog Management**

**Description and Priority:**

* Librarians can add, update, and remove books from the catalog.
* Priority: **High**

**Stimulus/Response Sequences:**

1. Librarian selects an option to add, update, or remove a book.
2. The system updates the catalog accordingly.

**Functional Requirements:**

* The system shall allow librarians to **add books**, including details such as title, author, and category.
* The system shall allow librarians to **update book details**.
* The system shall allow librarians to **remove books** from the catalog.
* The system shall update the catalog automatically when a book is added, removed, or modified.

**3.3 Book Search & Catalog Access**

**Description and Priority:**

* Users can search for books by title, author, or category.
* Priority: **High**

**Stimulus/Response Sequences:**

1. User enters a search query.
2. The system fetches and displays relevant book details.

**Functional Requirements:**

* The system shall allow users to **search for books** using filters.
* The system shall display **book availability status**.
* The system shall provide book **category-wise browsing**.

**3.4 Book Borrowing & Returning**

**Description and Priority:**

* Users can borrow and return books, with due dates assigned.
* Priority: **High**

**Stimulus/Response Sequences:**

1. User selects a book to borrow.
2. System checks availability and issues the book.
3. User returns the book, and the system updates the status.

**Functional Requirements:**

* The system shall allow users to **borrow books** and assign a due date.
* The system shall track **due dates** and **return status**.
* The system shall notify users upon successful return.

**3.5 Book Reservation & Renewal**

**Description and Priority:**

* Users can reserve books that are currently unavailable and renew books before the due date.
* Priority: **Medium**

**Stimulus/Response Sequences:**

1. User reserves a book.
2. The system notifies them when the book is available.

**Functional Requirements:**

* The system shall allow users to **reserve books**.
* The system shall allow users to **renew borrowed books** before the due date.

**3.6 Overdue Notifications & Fine Management**

**Description and Priority:**

* The system notifies users about overdue books and calculates fines.
* Priority: **Medium**

**Stimulus/Response Sequences:**

1. System calculates overdue fines based on delay days.
2. System sends notifications via email/SMS.

**Functional Requirements:**

* The system shall calculate **fines for overdue books**.
* The system shall send **reminders for due dates**.
* The system shall allow **fine payments online**.

**3.7 Book Issuing**

**Description and Priority:**

* Librarians issue books to users.
* Priority: **High**

**Stimulus/Response Sequences:**

1. Librarian selects a user and book details.
2. System verifies the borrowing limit.
3. System updates the book status as issued.

**Functional Requirements:**

* The system shall allow librarians to **issue books** to users.
* The system shall update the book’s **availability status**.

**3.8 Reporting & Analytics**

**Description and Priority:**

* Librarians can generate reports on book transactions.
* Priority: **Medium**

**Stimulus/Response Sequences:**

1. Librarian selects a report type.
2. System generates and displays the report.

**Functional Requirements:**

* The system shall generate reports for **issued books, overdue books, and fines**.
* The system shall allow exporting reports in **PDF and Excel formats**.

**4. External Interface Requirements**

**4.1 User Interfaces**

* The system shall have a **web-based dashboard** for users and librarians.
* The system shall provide an **intuitive UI for book searches and transactions**.

**4.2 Hardware Interfaces**

* The system shall be accessible via **PCs, tablets, and mobile devices**.
* The system shall support **barcode scanners** for book tracking.

**4.3 Software Interfaces**

* The system shall integrate with **institutional student portals** for authentication.
* The system shall connect to **a database for book storage and tracking**.

**4.4 Communication Interfaces**

* The system shall support **email/SMS notifications** for due dates and reservations.
* The system shall support **secure encryption for user data**.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The system should be able to support up to 1000 concurrent users without performance degradation.
* The response time for retrieving book details should not exceed 2 seconds under normal load conditions.
* The system should be able to process 50 book issue/return transactions per minute.
* Daily data backup should be completed within 30 minutes.
* The system should be accessible 99.9% of the time, ensuring minimal downtime.

**5.2 Safety Requirements**

* The system should prevent data loss by implementing automated daily backups.
* Users should not be able to modify or delete critical library records without proper authorization.
* The system should have an auto-logout feature after 10 minutes of inactivity to prevent unauthorized access.
* The system should comply with relevant data protection regulations (e.g., GDPR for libraries handling personal information).

**5.3 Security Requirements**

* Users must authenticate using a unique username and strong password.
* Role-based access control (RBAC) should be implemented to ensure only authorized users can modify library records.
* All sensitive data (e.g., user credentials, transaction logs) must be encrypted using AES-256.
* The system should support multi-factor authentication (MFA) for administrator accounts.
* The system should maintain an audit log for tracking all administrative actions and sensitive data access.

**5.4 Software Quality Attributes**

* **Availability:** The system should be available 24/7 except during scheduled maintenance.
* **Usability:** The user interface should be intuitive and accessible to non-technical users.
* **Scalability:** The system should be capable of expanding to accommodate a larger collection and more users without significant rework.
* **Maintainability:** The system should have modular architecture to facilitate easy updates and debugging.
* **Interoperability:** The LMS should integrate seamlessly with external databases and third-party library services.
* **Testability:** The system should support automated and manual testing for all major functionalities.

**6. Other Requirements**

* **Database Requirements:** The LMS should use a relational database (e.g., MySQL, PostgreSQL) for storing library records.
* **Legal Requirements:** The system should comply with local library regulations and copyright laws.
* **Internationalization:** The system should support multiple languages for diverse user accessibility.
* **Data Retention:** User and transaction records should be retained for at least 5 years before archiving.

**Appendix A: Glossary**

* **LMS:** Library Management System
* **RBAC:** Role-Based Access Control
* **MFA:** Multi-Factor Authentication
* **AES-256:** Advanced Encryption Standard (256-bit)

**Appendix B: Analysis Models**

(Include relevant diagrams such as Data Flow Diagrams (DFD), Entity-Relationship Diagrams (ERD), and Use Case Diagrams.)

**Appendix C: Issues List**

* **TBD:** Finalize the choice of database system.
* **Pending Decision:** Determine the exact duration for auto-logout.
* **Information Needed:** Compliance requirements for specific geographical locations.
* **Conflict Resolution:** Deciding between cloud-based and on-premise LMS deployment.