Practical No:02

Aim: Prepare broad SRS (Software Specification requirement) for the selected problem statement.

Project Title: Library Management System
Date:
Prepared by:

1. INTRODUCTION

1.1 Purpose

This document outlines the software requirements for the Library Management System (LMS). The intended audience includes developers, testers, project managers, and end-users such as librarians and students.

1.2 Scope

The Library Management System will automate the process of managing books, users, and borrowing activities in an educational institution or public library. The system will allow users to search, borrow, and return books, while administrators can manage inventory and track transactions.

Objectives:

- Efficient book cataloging and tracking
- User management for students and librarians
- Improved reporting and auditing of transactions
- Online access to library resources

Benefits:

- Saves time and effort in managing library operations
- Minimizes manual errors
- Increases user convenience with digital access

1.3 Definitions, Acronyms, and Abbreviations

- **LMS** – Library Management System
- **UI** – User Interface
- **DB** – Database
- **ISBN** – International Standard Book Number
- **CRUD** – Create, Read, Update, Delete

1.4 References

- IEEE SRS Standard (IEEE 830-1998)
- https://www.librarything.com
- Internal Library Policy Document, 2023

1.5 Overview

This SRS is structured to first present an overall description of the system, followed by specific requirements including functional, non-functional, and interface requirements. Appendices include diagrams and supporting material.

2. OVERALL DESCRIPTION

2.1 Product Perspective

The LMS is a standalone web-based system but can be integrated with existing student information systems for user synchronization.

2.2 Product Functions

- Book cataloging and inventory management
- User registration and authentication
- Search and filter books
- Book issue and return with due date tracking
- Fine calculation for overdue books
- Administrative dashboard for reporting and analytics

2.3 User Classes and Characteristics

- **Librarian/Admin:** Full access to manage books, users, and transactions.
- **Student/User:** Can search, view, borrow, and return books.
- **Guest:** Can browse catalog with limited access.

2.4 Operating Environment

- **Hardware:** Standard PC or laptop
- **Software:** Web browser, Windows/Linux OS
- **Network:** Internet/Intranet access
- **Database:** MySQL/PostgreSQL
- **Backend:** PHP/Node.js/Python
- **Frontend:** HTML, CSS, JavaScript

2.5 Design and Implementation Constraints

- Developed using open-source technologies
- Mobile responsiveness required
- Database must support relational operations
- Must comply with institutional security policies

2.6 User Documentation

- User Manual (PDF/Online)
- Admin Guide
- FAQ and Troubleshooting section
- On-screen tooltips and help prompts

2.7 Assumptions and Dependencies

- Internet access is available for all users
- Server and database hosting provided by the institution
- Users are familiar with basic web navigation
- Barcode scanner support assumed for future enhancements

3. Specific Requirements

3.1 Functional Requirements

- **FR1:** The system shall allow users to register with a valid student ID and email.
- **FR2:** The system shall authenticate users with a password.
- **FR3:** The user shall be able to search for books using title, author, or ISBN.
- **FR4:** The system shall allow users to borrow and return books.
- **FR5:** The system shall calculate and display fines for overdue books.
- **FR6:** The librarian shall be able to add, update, and delete books from inventory.
- **FR7:** The system shall generate reports on issued books and user activity.
- **FR8:** The system shall notify users of due dates via email.
- **FR9:** The system shall prevent a book from being issued if no copies are available.
- **FR10:** The admin shall be able to manage user accounts and permissions.

3.2 Non-functional Requirements

- **Performance:** System should respond to user requests within 2 seconds.
- **Security:** Passwords must be hashed; access to admin features must be restricted.
- **Usability:** UI should be intuitive with consistent navigation.
- **Reliability:** 99.5% uptime required; regular backups must be scheduled.
- **Maintainability:** Code should follow modular structure and documented comments.
- **Portability:** System should run on any modern browser and OS.

3.3 Interface Requirements

User Interfaces

- Login page
- User dashboard
- Book search and results page
- Book detail view
- Admin dashboard with CRUD operations

Hardware Interfaces

- Barcode reader support (optional/future use)

Software Interfaces

- MySQL or PostgreSQL database
- Email service API (e.g., SMTP or SendGrid)

Communication Interfaces

- HTTP/HTTPS protocol
- RESTful APIs for future integration

4. Appendices

4.1 Diagrams

- Use Case Diagram
- Entity-Relationship (ER) Diagram
- Data Flow Diagram (DFD)
- Class Diagram

Conclusion: