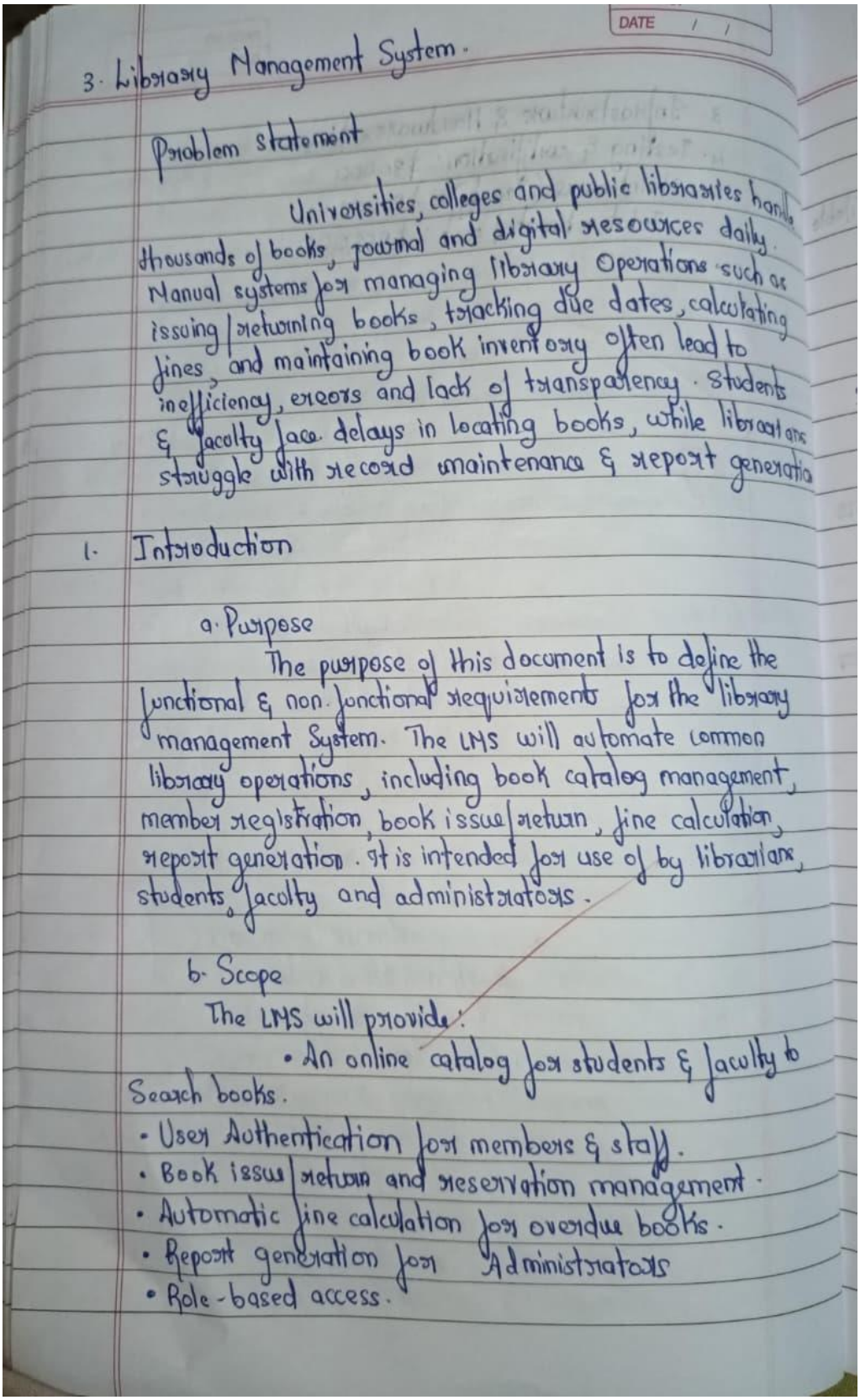
OOM LAB

SRS Document

Library Management System



3. Library Management System.

Problem statement

Universities, colleges and public libraries handle thousands of books, journal and digital resources daily. Manual systems for managing library operations such as issuing/returning books, tracking due dates, calculating fines, and maintaining book inventory often lead to inefficiency, errors and lack of transparency. Students & faculty face delays in locating books, while librarians struggle with record maintenance & report generatio

1. Introduction

a. Purpose

The purpose of this document is to define the functional & non-functional requirements for the library management System. The LMS will automate common library operations, including book catalog management, member registration, book issue/return, fine calculation, report generation. It is intended for use of by librarians, students, faculty and administrators.

b. Scope

The LMS will provide:

- An online catalog for students & faculty to Search books.
- User Authentication for members & staff.
- Book issue/return and reservation management.
- Automatic fine calculation for overdue books.
- Report generation for Administrators
- Role-based access.

PAGE NO.
DATE / /

c. Overview

- The system will be implemented as a web-based application with a relational database backend, Users interact via a browser or mobile device. The system ensures secure login accurate record keeping and scalability for growing library needs.

2. General Description

- Users: librarians, students, faculty, Admin
- System Environment: Web-based Interface with SQL database
- Assumptions: Users have Internet access, unique credentials and basic computer Knowledge.
- Dependencies: stable Internet, functioning server and secure authentication system

3. Functional Requirements.

a. User Management
- Register/login/logout
- Role based access

b. Book Management
- Add, update and delete books.
- Maintain metadata.

c. Search & Browse
- search books by title, author, genre or ISBN
- view book availability.

d. Issue & Return
- issue books to members
- Record return dates
- Auto calculate fines.

PAGE NO.
DATE / /

e. Reservation
- Reserve Unavailable books.
- Notify users when reserved books are available

d. Reports & logs
- Daily/Monthly transaction reports
- Inventory reports.

4. Interface Requirements.
- User Interface:
  - web-based dashboard for admins/librarians
  - Mobile-friendly member portal
- Hardware Interface:
  - server with atleast 8GB RAM, 4 core processor
  - Barcode Integration Scanner.
- Software Interface:
  - Database: MySQL
  - Frontend: HTML, CSS, JavaScript
  - Backend: Java/Node.Js

5. Performance Requirements
- System must handle 1000+ concurrent users
- Search results must appear within 2 secs
- Support database of 1000000 + books

6. Design Constraints
- Must comply with data d Security standards
- System should be platform-Independent
- Must support backup & recovery in case of System failure.

PAGE NO.
DATE / /

7. Non-functional Attributes
   - Reliability, usability, maintainability, Security, Scalability.

8. Preliminary Schedule & Budget

Schedule
- Requirement Analysis & Design: 3 weeks
- Database Design & setup: 2 weeks
- Frontend & Backend Development: 5 weeks.
- Testing & QA: 3 weeks
- Deployment & training: 2 weeks

Budget
- Deployment cost: $15000
- server hosting & maintanance: $2000/yr
- Licensing & tools: $1000
- Training & support: $1500

total estimate: $20,000