

# **Software Design Specifications**

**for**

## **Library Management System**

**Prepared By:**

AI053 (G. Chaithri),

AI020 (Anvitha Reddy),

AI133 (P. Pravallika)

AI101 (M. Sowmitha)

AI202 (P. Vandana)

AI009 (Abhi Aitha)

## Document Information

Title	Library Management System
Project Manager	Murali Krishna/ Avinash Arun Chauhan
Document Version No.	1.0
Document Version Date	08-Apr-2025
Prepared By	AI053, AI020, AI101, AI133
Preparation Date	06-Apr-2025

## Version History

Ver. No	Ver. Date	Description	Filename
1.0	08-Apr-2025	Initial Draft Submission	Library_Management_SDS_v1.doc

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present the design specification of the Library Management System (LMS) developed for college use. It details the software architecture, design elements, and behavior of the system to assist developers, maintainers, and stakeholders.

## 1.2 Scope

This system manages book inventory, student and staff users, borrowing/returning of books, barcode-based book entry, and a unique feature: online book reservation. The design ensures user-friendly operations.

## 1.3 Definitions, Acronyms, and Abbreviations

- LMS: Library Management System
- UI: User Interface
- ISBN: International Standard Book Number
- DB: Database

## 1.4 References

- Project Guidelines Document provided by Mahindra University
- Java and html documentation

## 2. Use Case View

### 2.1 Use Case

- Borrow Book – Student scans a book's barcode, system records issue.
- Return Book – Return entry made; system calculates fines if due.
- Reserve Book – Student reserves an unavailable book; notified when available.
- Add New Book – Admin scans and adds book using barcode.
- Search Book – Student can search for books based on title, author, year published, and ISBN.

## 3. Design Overview

### 3.1 Design Goals and Constraints

- Integration with barcode scanner hardware.
- Smooth performance for up to 100 simultaneous users.
- Responsive web UI using Django or JavaFX (if desktop).
- Real-time notification of reserved book availability.

### 3.2 Design Assumptions

- All users have valid credentials from the college system.

### 3.3 Significant Design Packages

- UserManagement: Handles login, registration.
- BookManagement: Book data and operations.
- ReservationSystem: Manages reservation queue.

### 3.4 Implemented Application External Interfaces

Interface Name	Module Implementing the Interface	Description
ReserveBookAPI	ReservationSystem	Allows users to reserve books

## 4. Logical View

### 4.1 Design Model

Includes classes: User, Book, Reservation, Transaction.

- User has roles: student, staff, admin.

- Book includes fields: title, author, ISBN, year published.
- Reservation tracks user-book relation when reserved.

## 4.2 Use Case Realization

For “Reserve Book”:

1. User logs in.
2. Searches for book.
3. If unavailable, clicks Reserve.
4. System logs request, adds to waiting list.
5. On return, notifies next user in queue.

## 5. Data View

### 5.1 Domain Model

Entities:

- User(id, name, role, email)
- Book(id, title, author, isbn, status)
- Reservation(id, user\_id, book\_id, date\_reserved)
- Transaction(id, user\_id, book\_id, issue\_date, return\_date, fine)

### 5.2 Data Model (Persistent Data View)

#### 5.2.1 Data Dictionary

- Book.status: ENUM(available, issued, reserved)
- Reservation.date\_reserved: TIMESTAMP

## 6. Exception Handling

- ReservationLimitExceededException – Max reservations reached.
- DatabaseConnectionError – Triggered when DB is unreachable.
- All exceptions are logged with timestamp in error\_log.txt.

## 7. Configurable Parameters

Parameter	Description	Dynamic
maxReservations	Max number of books a user can reserve	Yes
finePerDay	Fine charged per late day	Yes

## **8. Quality of Service**

### **8.1 Availability**

- 99% uptime expected; critical services (like reservation) prioritized.

### **8.2 Security and Authorization**

- Role-based access: Admin, Staff, Student.
- Secure login, encrypted passwords.

### **8.3 Load and Performance Implications**

- Optimized DB queries.
- Local caching for common book searches.

### **8.4 Monitoring and Control**

- Daily logs, admin dashboard with activity stats.