Software Design Specification

LIBRARY MANAGEMENT SYSTEM

Internal Advisor:

- Librarians
- Staff
- Administrator
- Project Manager
- Developers

External Advisor:

- Software Consultants
- Suppliers
- Library users
- Students
- Government Bodies
- Maintenance

Project Team:

- **Saba Maryam**
- **Zenab Noor**
- **Shahzeen Fatima**

Submission Date:

Document Information

Category	Information
Customer	UOS – Department of Computer Science
Project	Library Management System
Document	Software Design Specification
Document Version	1.0
Identifier	SDS_Project_Group_03_LMS
Status	Draft
Author(s)	Saba Maryam, Shahzeen Fatima, Zenab Noor
Approver(s)	Saba Maryam, Shahzeen Fatima, Zenab Noor
Issue Date	November 19, 2025
Document Location	Stored in project GitHub repository
Distribution	1. Dr. Illyas 2. Saba Maryam, Shahzeen Fatima, Zenab Noor 3. UOS – Department of Computer Science

Definition of Terms, Acronyms and Abbreviations

Term	Description
SDS	Software Design Specification
UOS	University Of Sargodha
LMS	Library Management System
UI	User Interface

Table of Contents

1. Introduction

1.1 Purpose of Document

This document provides the detailed software design specifications for the Library Management System. It is intended for developers, testers, and other project stakeholders. The project will use ***"Object-Oriented Design methodology"*** *to ensure modularity and reusability.*

1.2 Project Overview

The Library Management System (LMS) is designed to handle automatically library operations such as book borrowing, returning, managing book records, and user registration. It will provide an easy-to-use interface for librarians and members to manage the resources efficiently. The software will be developed using ***"Object-Oriented Design"****, focusing on modularity, scalability, and maintainability.*

1.3 Scope

- The system will manage book records, including adding, updating, and deleting book records.
- It will handle member registration, issue, and return of books.
- It will check due dates and calculate fines for late returns.
- It will generate reports on book availability and member activity.
- Users will be able to search for books and check their availability.

Out of Scope:

- The system will not handle digital books or online reading material.
- It will not give smart suggestions or use artificial intelligence.
- It won't connect with other libraries.
- The system is limited to managing the internal operations of a single library environment.

2. Design Considerations

- *The system should be user-friendly.*
- *The system should be secure.*
- *The system should be compatible with existing library computers.*
- *It must handle book records, user roles, borrowing and returning books properly.*
- *The software should also be flexible so future updates can be added easily.*
- *The software should ensure data protection and easy maintenance.*

2.1 Assumptions and Dependencies

- *Security will be built into the design to keep user information safe.*
- *The system will be designed so it's easy to update and fix if needed.*
- *The design will focus on making the system work quickly and efficiently.*
- *The design must be compatible with existing hardware and software.*
- *Design allows easy growth for more users and data.*

2.2 Risks and Volatile Areas

- *Changes in user requirements may affect design.*
- *New technology may need updates in the system.*
- *Security issues could require quick fixes.*
- *Design will be flexible to handle changes smoothly.*
- *Backup plans will be ready for unexpected problems.*
- *Integration issues with other systems may occur.*
- *Hardware or software limitations might impact design choices.*
- *Changes in rules or policies can affect system features.*
- *Performance issues may arise with more users or data.*

3. System Architecture

The system is divided into separate components for better organization and functionality.
- Key components include:

- *User Interface (**UI**): Allows users to interact with the system easily.*
- *Functional Modules:*
  - *Book Management*
  - *Member Management*
  - *Issue/Return Module*
- *Database Layer: Stores and manages data like books, members, and transactions.*

Each component has its own responsibility but works together with others. The design ensures smooth communication between modules for efficient performance.

3.1 System Level Architecture

- *The system is split into main parts: user interface, logic, and database.*
- *These parts work together through clear connections.*
- *It can connect with other tools like barcode scanners if needed.*
- *All parts will run on the same computer or local network.*

. *Errors will be handled in one place, and data will be checked properly.*

3.2 Sub-System / Component / Module Level Architecture

The Library Management System is divided into the following key modules:

- ***User Management Module***
  Handles librarian and member registration, login, and roles.
- ***Book Management Module***
  Manages book records, including add, update, delete, and search.
- ***Issue/Return Module***
  Tracks issued books, return dates, and fines.
- ***Inventory Module***
  Maintains stock levels of physical books.
- ***Report Module***
  Generates reports such as borrowed books, due dates, and user activity.

- ***Database Module***
  Stores all system data and ensures secure access.

3.3 Sub-Component / Sub-Module Level Architecture (1…n)

1. User Management Module

- *Registration Handler (Manages user sign-up)*
- *Login Validator (Handles authentication)*
- *Role Manager (Assigns roles (Admin, Librarian, Member))*

2. Book Management Module

- *Book Entry (Add/update book details)*
- *Book Search (Search by title, author, or category)*
- *Book Deletion (Remove outdated records)*

3. Issue/Return Module

- *Issue Handler (Issues books to members)*
- *Return Processor (Handles book return)*
- *Fine Calculator (Calculates late return fines)*

4. Report Module

- *Borrowed Report (List of currently issued books)*
- *Due Report (Books that are due or late)*
- *Activity Report (User-wise activity logs)*

5. Inventory Module

- *Stock Tracker (Monitors number of copies available)*
- *New Stock Entry (Adds new books to inventory)*
- *Damaged/Lost Handler (Updates inventory for lost or damaged books)*
- *Reorder Alert (Notifies when stock is low (optional feature))*

6. Database Module

- *Data Connection Manager (Connects system to database)*
- *Query Processor (Handles queries (insert, update, fetch))*
- *Backup Handler (Manages data backup and recovery)*
- *Data Integrity Checker (Ensures consistency and validity of data)*

4. Design Strategies

Modular Design: The system is divided into separate modules (like user management, inventory, database) to improve maintainability and scalability.

Object-Oriented Approach: The design follows OOP principles like encapsulation, inheritance, and polymorphism to promote code reusability.

Layered Architecture: The system is built in layer, one for user interface, one for logic, and one for data to keep things clear and separate.

Use of Relational Database: A proper database (like MySQL) will store information about books, users, and issued books.

Error Handling Strategy: The system will include proper methods to deal with errors so it works safely and smoothly.

Security Considerations: User login and access control will be used to protect data and keep the system secure.

4.1 Strategy 1…n

- ***Future Extension***
  Modular design allows easy addition of features like e-book support but needs more planning.
- ***System Reuse***
  OOP enables reuse of features like login but slightly complex structure.
- ***User Interface***
  Clean and simple UI for easy use but may limit advanced features.
- ***Data Management***
  MySQL ensures secure and reliable data storage but needs proper setup.
- ***Concurrency***
  Supports multiple users at once but harder to manage data access.

5. Detailed System Design

A detailed design should include the following:

- *Detailed class diagram along with a detailed description of all attributes, functions or methods specifying interactions between different classes/modules.*
- *Detailed Sequence diagram with parameter list*
- *State Transition Diagram*
- *Logical data model (E/R model)*
- *Physical data models*
- *Detailed GUIs*

6. References

This section should provide a complete list of all documents referenced at specific point in time. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained (This section is like the bibliography in a published book).

Ref. No.	Document Title	Date of Release/ Publication	Document Source

Ref. No.	Document Title	Date of Release/ Publication	Document Source
PGBH01-2025-Proposal	Project Proposal	Oct 20, 2025	<Give the path of your Project repository/Folder>
PGBH01-2025-FS	Functional Specification	Oct 20, 2025	<Give the path of your Project repository/Folder>

7. Appendices

Include supporting detail that would be too distracting to include in the main body of the document.