Library Management System

Rachel Watts

Final Project

System Design

Jafrina Jabin

12/10/2025

Table of Contents

1. Introduction

The purpose of this document is to describe the requirements, design, and implementation of the Library Management System developed as a final project. This system was created to replace manual and spreadsheet-based library tracking methods with an automated, reliable software solution. The document consolidates all prior submissions and reflects the complete system design and development process.

2. Customer Problem Statement

Many libraries still rely on paper records or spreadsheets to track borrowed books and manage inventory. These methods are inefficient, error-prone, and time-consuming for librarians and library members. Manual systems make it difficult to quickly determine book availability, track overdue items, and maintain accurate records.

The Library Management System addresses these issues by providing a centralized system that automates book tracking, borrowing, returning, and inventory management. This allows librarians to operate more efficiently and members to quickly access accurate book availability information.

3. System Requirements

3.1 Stakeholders

- Library Members
- Librarians
- Library Manager / Administrator
- System Developer

3.2 System Goals

- Provide an easy way to search for books
- Track borrowed and returned books
- Reduce manual record-keeping
- Improve accuracy and efficiency

4. Functional Requirements Specification

FR1	**High**	**Members can search for books by title or author**
FR2	**High**	**System displays availability status of books**
FR3	**High**	**Members can borrow books**
FR4	**High**	**Members can return books**
FR5	**Medium**	**Librarians can add or remove books**
FR6	**Medium**	**Librarians can edit book information**
FR7	**Medium**	**System tracks overdue books and fines**
FR8	**Medium**	**System updates availability automatically**

5. Use Cases

Primary Actors

- Library Member
- Librarian

Use Case Examples

- Member searches for a book
- Member borrows a book
- Member returns a book
- Librarian adds or removes a book
- Librarian views overdue books

Each use case defines the interaction between users and the system to accomplish a specific goal.

6. System Sequence Diagrams

Borrow a Book

1. Member logs into the system
2. Member searches for a book
3. System checks availability
4. System updates book status and assigns due date

Return a Book

1. Member selects a book to return
2. System updates availability
3. System checks overdue status
4. System displays confirmation and fines if applicable

7. Activity Diagram

The activity flow of the system begins when a user logs in and selects an action such as searching, borrowing, or returning a book. The system processes the request, updates records automatically, and returns feedback to the user. This cycle continues until the user exits the system.

8. User Interface Specification

The system uses a console-based interface.

Main Screens

- Login Screen
- Search Screen
- Borrow / Return Screen
- Librarian Dashboard

Menus guide users through actions using numbered options, minimizing user effort and input errors.

9. Traceability Matrix

FR1	**Search for books**
FR2	**Check availability**
FR3	**Borrow book**
FR4	**Return book**
FR5	**Add / Remove books**
FR6	**Librarians can edit book information**
FR7	**System tracks overdue books and fines**
FR8	**System updates availability automatically**

This matrix ensures all requirements are implemented and tested through defined use cases.

10. System Architecture and System Design

The Library Management System follows a layered architecture:

- Presentation Layer: Console-based user interface
- Application Layer: Business logic and system control
- Data Layer: In-memory storage

This design separates concerns and supports scalability.

11. Algorithms and Data Structures

The system primarily uses basic data structures such as lists and objects to store books, members, and transactions. No advanced algorithms are required due to the scope of the system. Object-oriented programming principles are applied throughout the design.

12. User Interface Design and Implementation

The user interface is implemented using console menus that guide users through actions step-by-step. Input validation ensures users enter acceptable values, reducing errors and improving usability.

13. Test Design

Testing was primarily conducted through manual test cases to verify:

- Book searching
- Borrowing and returning functionality
- Automatic availability updates
- Overdue detection

Future versions of the system could incorporate automated testing.

14. Project Plan

Week	**Activity**
1-2	**Project setup and planning**
3-4	**Login and user roles**
5-7	**Book management features**
8	**Midterm demo**
9-11	**Borrow/return features**
12-14	**Testing and improvements**
15	**Final demo and documentation**