Software Requirements Specification (SRS)

Library Management System

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the Library Management System (LMS). The LMS will enable efficient management of library operations, including cataloging, lending, and tracking resources.

1.2 Scope

The LMS is intended to streamline library management processes by automating routine tasks such as book lending, returning, inventory management, and reporting. It will be used by librarians, library members, and administrators to access and manage library resources.

1.3 Definitions, Acronyms, and Abbreviations

- **LMS**: Library Management System
- **ISBN**: International Standard Book Number
- **GUI**: Graphical User Interface

1.4 References

1. lEEE Standard 830-1998: Recommended Practice for Software Requirements Specifications.
2. Library Management Best Practices.

1.5 Overview

This document covers the system’s functional and non-functional requirements, assumptions, constraints, and use cases.

---

2. Overall Description

2.1 Product Perspective

The LMS is a standalone application designed to integrate with existing library hardware (e.g., barcode scanners) and support multiple user roles.

2.2 Product Functions

- Book cataloging (adding, updating, and deleting book records).
- User account management (creating and maintaining member profiles).
- Lending and returning books.
- Overdue notifications and fines.
- Reporting and analytics.

2.3 User Characteristics

- **Librarians**: Manage resources and user accounts.
- **Members**: Borrow and return resources.
- **Administrators**: Oversee system configuration and reporting.

2.4 Constraints

- The system must comply with local data protection regulations.
- It must support a minimum of 500 concurrent users.

2.5 Assumptions and Dependencies

- Users must have internet access to use the system remotely.
- Hardware dependencies include barcode scanners and printers.

---

3. Specific Requirements

3.1 Functional Requirements

1. **Book Management**
    - Add new books to the catalog.
    - Edit or remove existing book records.
    - Search for books using filters like title, author, genre, or ISBN.
2. **User Management**
    - Register new members.
    - Update member details.
    - Deactivate accounts as needed.
3. **Lending and Returning**
    - Record book borrowings and returns.
    - Generate overdue fines.
    - Notify users of overdue items via email or SMS.
4. **Reporting**
    - Generate usage statistics (e.g., most borrowed books).
    - Produce financial reports for fines collected.

3.2 Non-Functional Requirements

- **Performance**: The system should process transactions within 2 seconds.
- **Scalability**: Must support future expansion to handle up to 10,000 users.
- **Security**: User authentication using two-factor authentication.
- **Availability**: 99.9% uptime.
- **Usability**: Intuitive and user-friendly GUI.

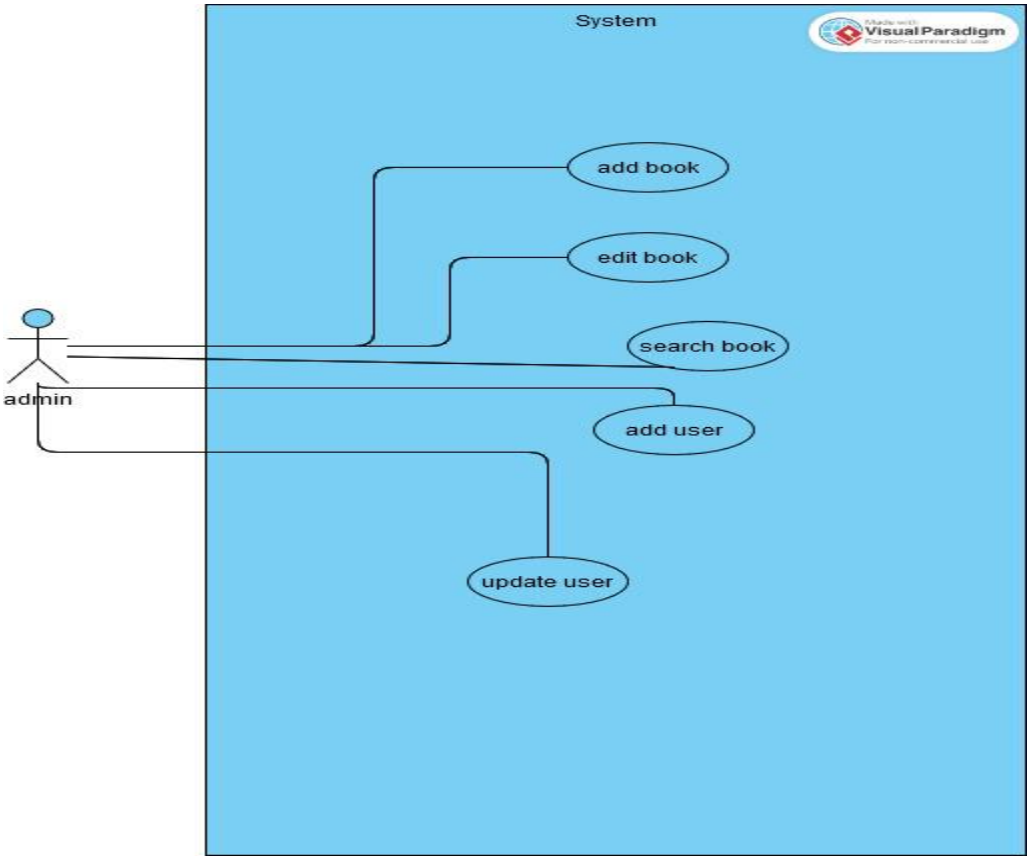
3.3 External Interface Requirements

- **User Interface**: Web-based application with responsive design.
- **Hardware Interfaces**: Barcode scanners, printers, and library card readers.
- **Software Interfaces**: Integration with email servers for notifications.

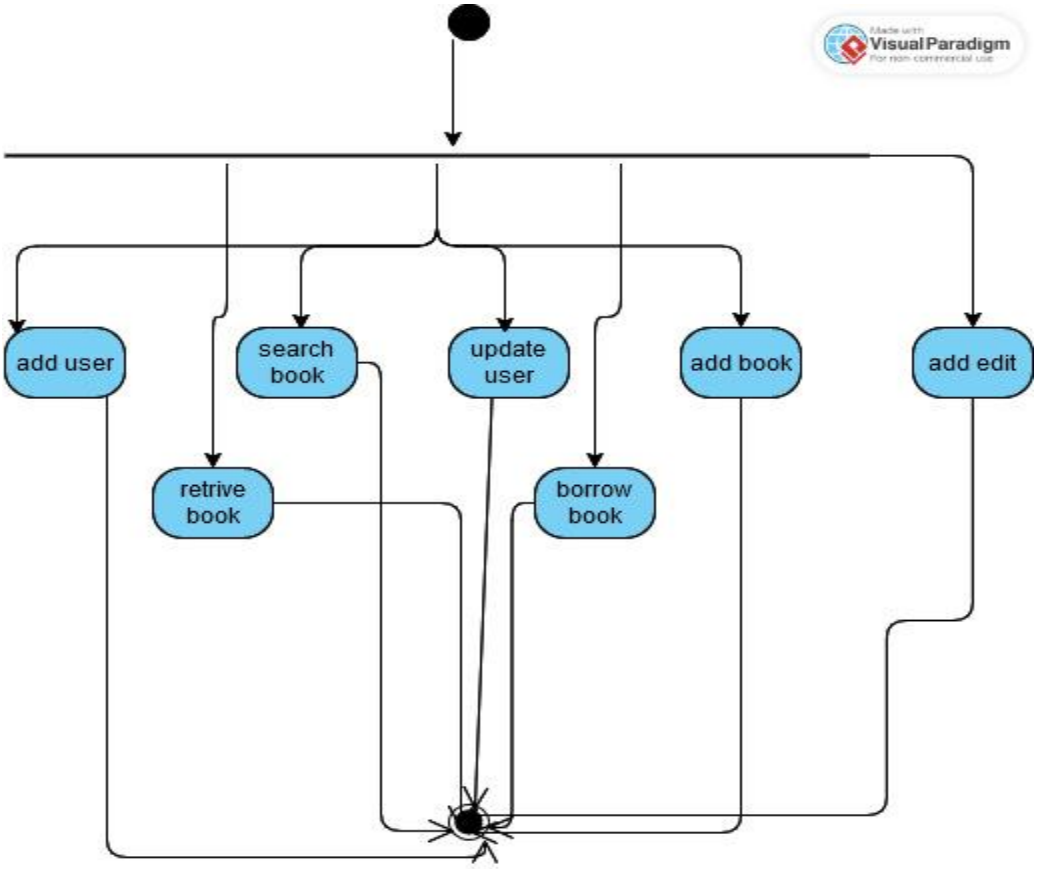
---

4. System Design

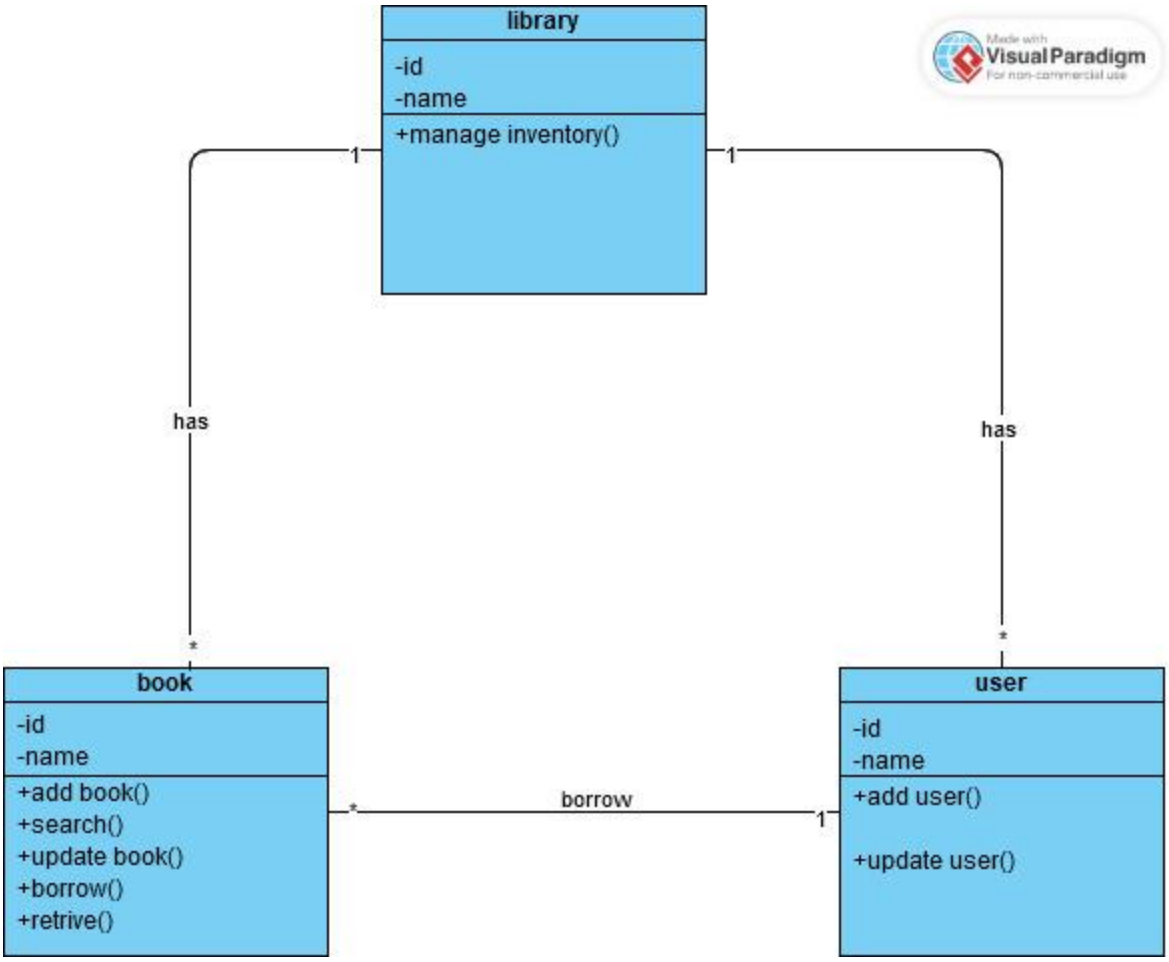
Use Case



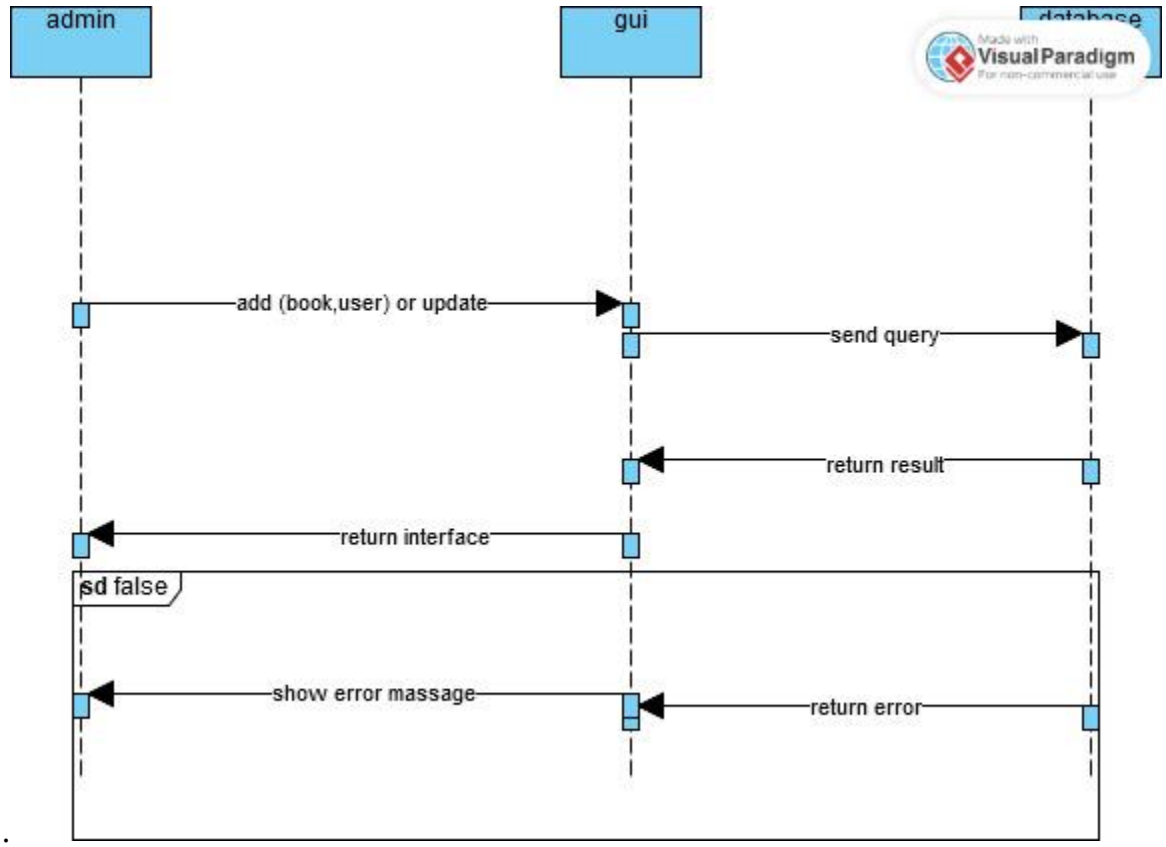
Activity Diagram



Class Diagram



Sequence Diagram


admin
gui
database
Made with VisualParadigm
For non-commercial use
add (book,user) or update
send query
return result
return interface
sd false
show error massage
return error
.

5. Testing

5.1 Unit Testing

- Verify individual components such as user registration, book management, and fine calculation.

5.2 Integration Testing

- Ensure seamless interaction between modules such as lending, returning, and notification systems.

5.3 System Testing

- Validate the overall functionality of the LMS under various scenarios, including high user load and data-intensive operations.

5.4 User Acceptance Testing

- Conduct testing with real users (librarians and members) to ensure the system meets user requirements and expectations.

5.5 Performance Testing

- Test the system’s response time and stability under peak usage conditions to ensure it meets the performance requirements.

---

6. Future Work

6.1 Advanced Search Features

- Implement semantic search capabilities to allow users to search by concepts and keywords rather than exact matches.

6.2 Mobile Application

- Develop a mobile app version of the LMS for Android and iOS platforms to enhance accessibility.

6.3 AI-Powered Recommendations

- Integrate AI algorithms to provide personalized book recommendations based on user preferences and borrowing history.

6.4 Multi-Language Support

- Extend the system to support multiple languages for improved usability by a diverse user base.

6.5 Integration with External Systems

- Enable integration with other library networks and third-party APIs for interlibrary loans and extended resource sharing.

---

7. Appendices

Appendix A: Glossary

- **Overdue**: Books not returned by the due date.
- **Fine**: Monetary penalty for overdue items.

Appendix B: Document History

- Version 1.0: Initial draft.