1. Introduction

1.1 Purpose

The Library Management System (LMS) aims to automate the management of books, members,

and transactions in a library, providing a user-friendly interface for librarians and library

members to perform their respective tasks efficiently.

1.2 Scope

The LMS will allow librarians to manage inventory, member information, and book transactions

(issue/return). Members can search for books, view their account details, and renew borrowed

books. The system will improve operational efficiency and reduce errors associated with manual

processes.

1.3 Definitions, Acronyms, and Abbreviations

- LMS: Library Management System
- Admin: Library administrator/librarian
- Member: Registered library user

1.4 References

Not applicable for this version.

2. Overall Description

2.1 System Overview

The system consists of two main components:

1. Admin Module: For librarians to manage books, members, and transactions.

2. Member Module: For users to search books and manage their library accounts.

2.2 Product Functions

- Admin Module:
- Add/update/delete books.
- Manage member accounts.
- Issue/return books.
- Generate reports on inventory and member activities.
- Member Module:
- Search for books by title, author, or category.
- View book availability.
- Renew borrowed books.
- View borrowing history.

2.3 User Classes and Characteristics

- Admin: Skilled in library management and system navigation.
- Member: Basic computer literacy to use the system's search and account functionalities.

2.4 Operating Environment

- Platform: Web-based system accessible via desktop or mobile devices.
- Browsers: Chrome, Firefox, Safari, Edge.

3. Requirements Specification

Functional Requirements

3.1 Admin Functions

1. Book Management:
    a. Add new books to the system.
    b. Update book details (e.g., title, author, quantity).
    c. Remove books from inventory.
2. Member Management:
    a. Add new members.
    b. Update member details.
    c. Remove inactive members.
3. Transaction Management:
4. Issue books to members.
5. Process book returns.
6. Track overdue books and apply fines.

3.2 Member Functions

a. Search Books:
b. Search by title, author, or genre.
c. Account Management:
d. View borrowing history.
e. Renew borrowed books (if eligible).

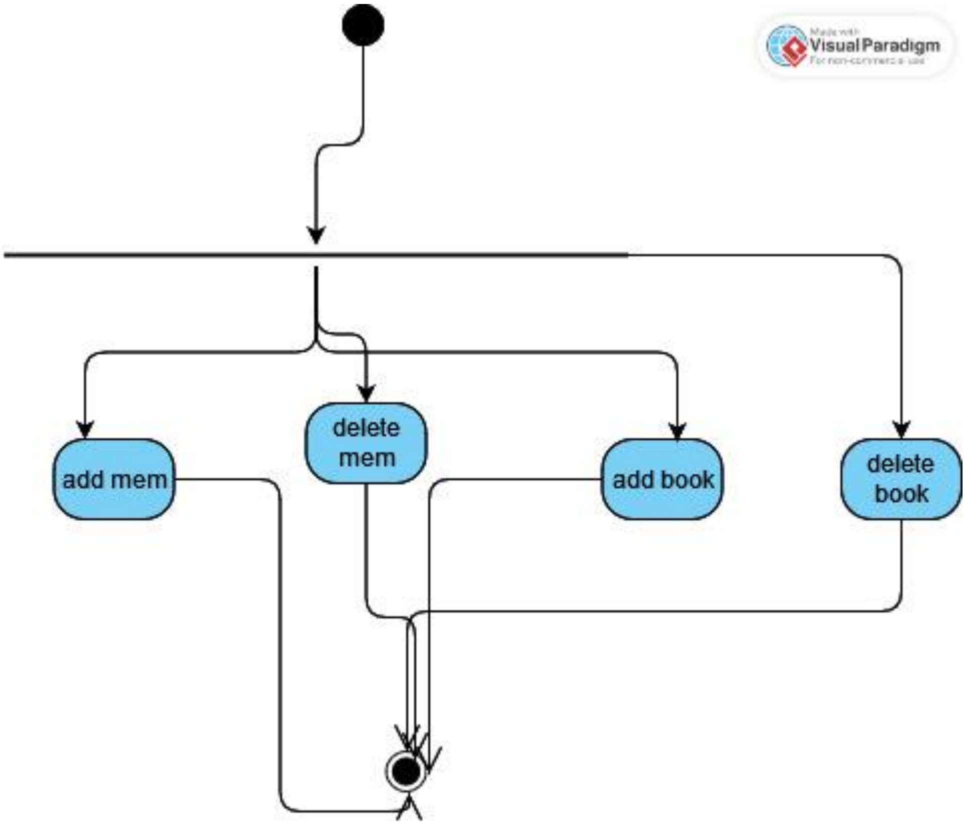
Non-Functional Requirements

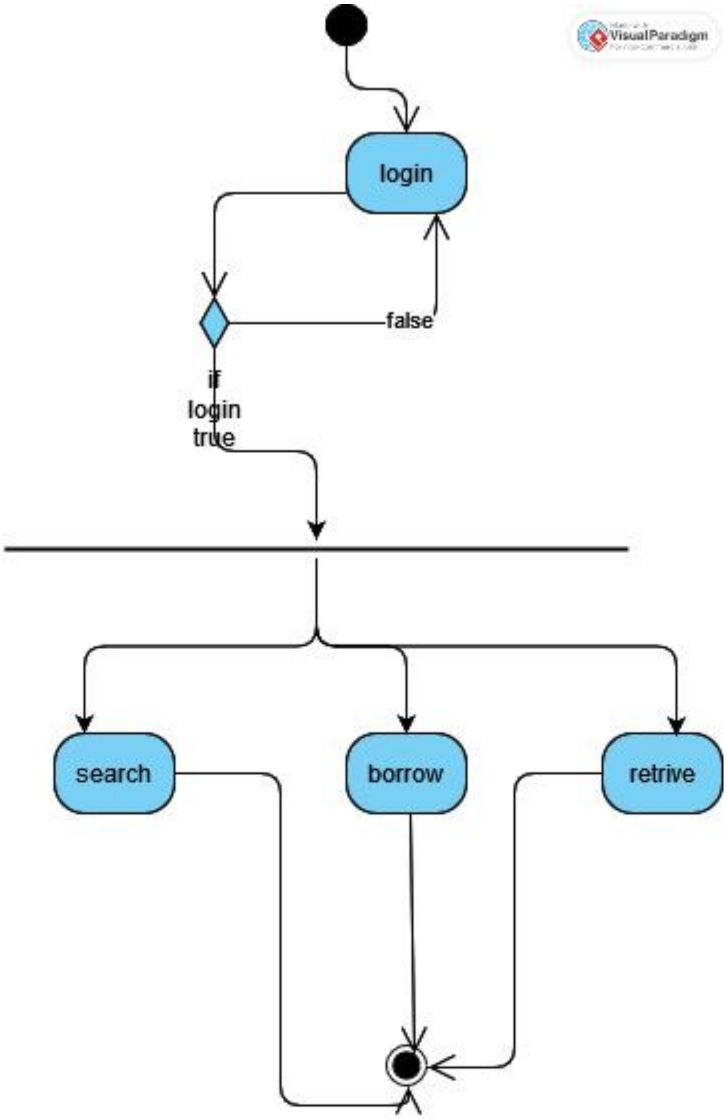
3.3 Performance Requirements

□ The system should handle up to 100 concurrent users.
□ Search queries should return results within 2 seconds.
□ 3.4 Security Requirements
□ Admin and member accounts must be protected by secure passwords.
□ All sensitive data (e.g., member information) must be encrypted.
□ 3.5 Usability Requirements
□ The interface should be intuitive and accessible for users with minimal
□ training.
o Scalability
□ The system should support the addition of new modules, such as e-book
□ management.

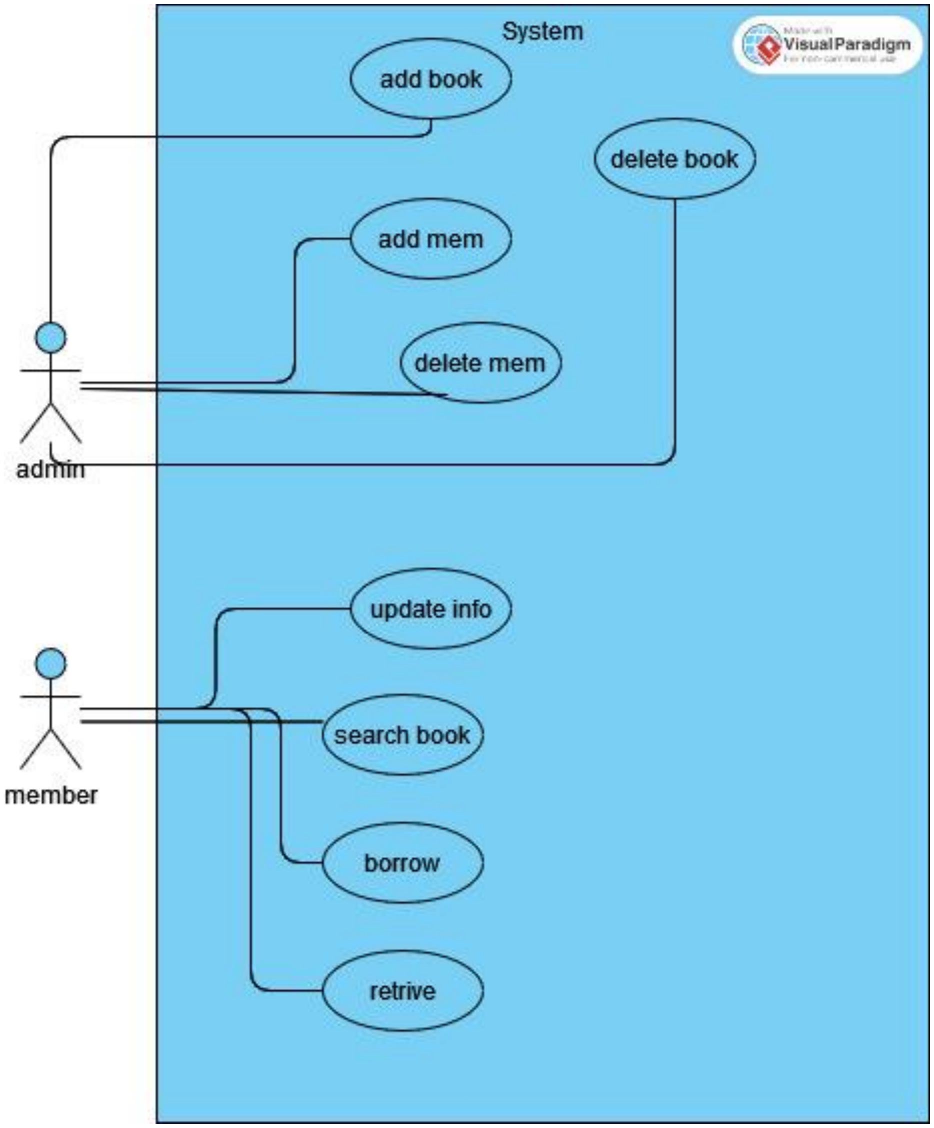
4. System Design

Activity Diagram

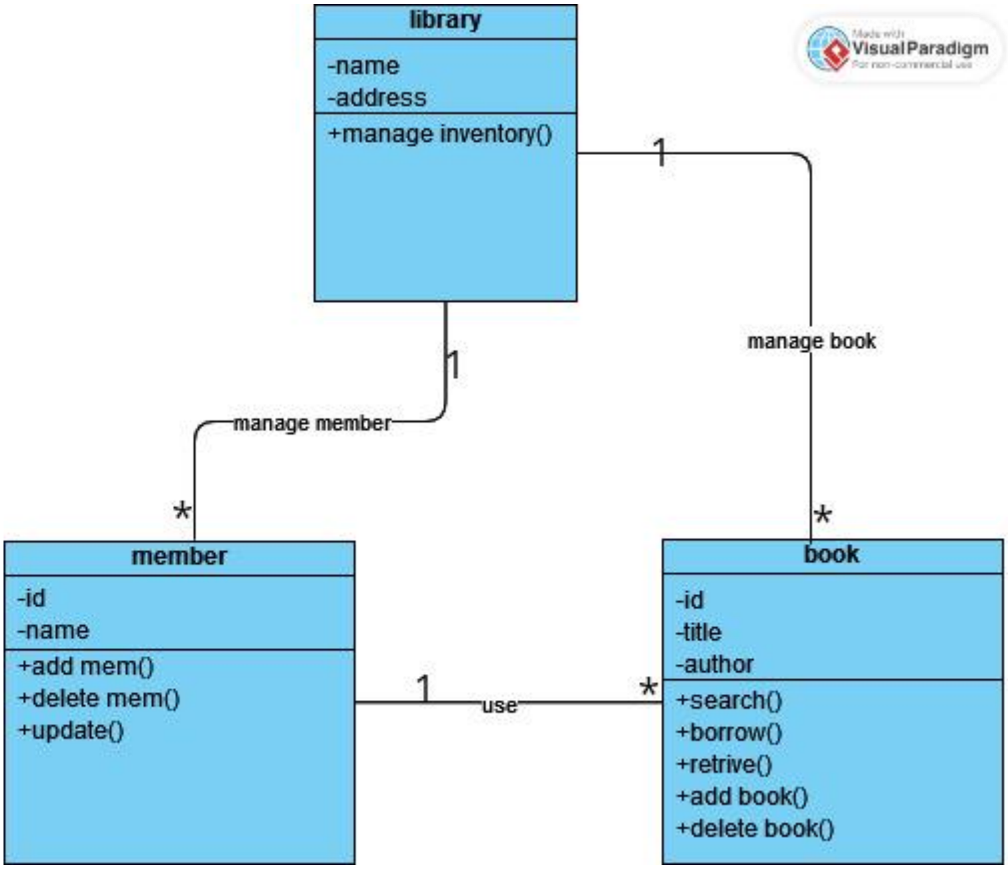

Made with Visual Paradigm For non-commercial use
add mem
delete mem
add book
delete book



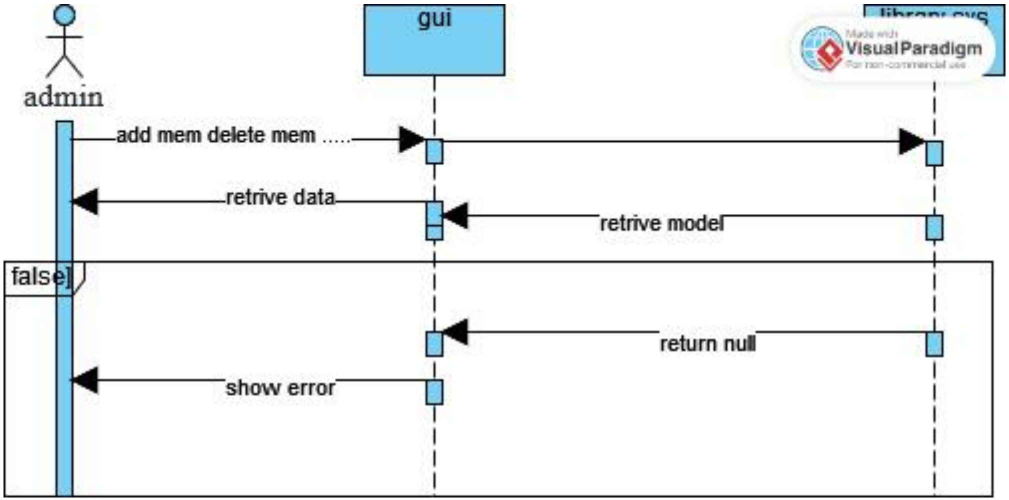
Use Case

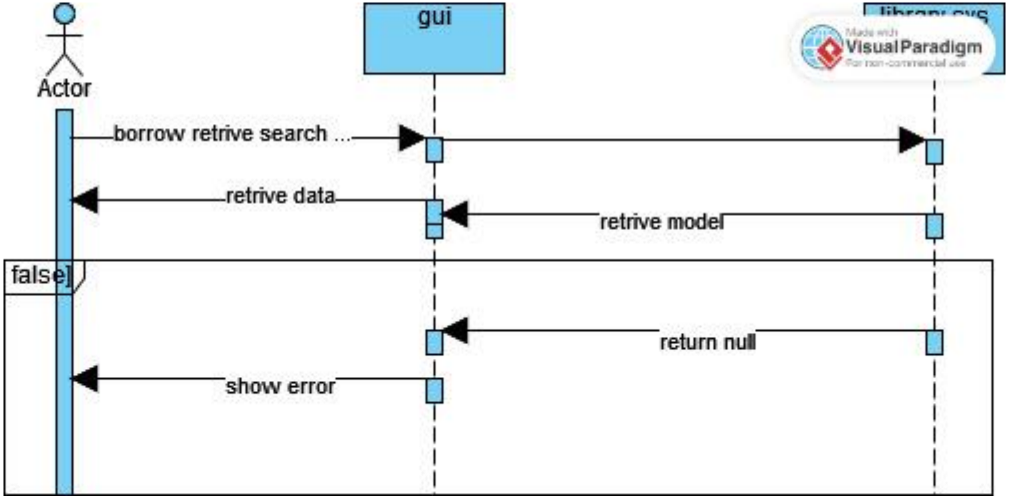


Class Diagram



Sequence Diagram


admin
gui
library sys
add mem delete mem .....
retrive data
retrive model
false]
return null
show error


Actor
gui
library sys
borrow retrive search ...
retrive data
retrive model
false]
return null
show error

5. Testing

Test Cases

1. Search Books
   - Input: Valid book title.
   - Expected Output: Display matching books.
2. Issue Books
   - Input: Valid memberID and bookID.
   - Expected Output: Record transaction and update book availability.

6. Maintenance and Support

- Data Backup: Schedule nightly database backups.
- Error Logging: Implement server-side logging for all API calls.
- Support: Contact the admin via email@example.com for technical support.

7. Future Enhancements

- Integration with external libraries for e-books.
- Multi-language support.
- Mobile application development.