KARNATAK LAW SOCIETY'S
GOGTE INSTITUTE OF TECHNOLOGY
UDYAMBAG, BELAGAVI-590008
(An Autonomous Institution under Visvesvaraya Technological University, Belgaum)
(APPROVED BY AICTE, NEW DELHI)

Course Activity Report on

"Library Management System"

submitted in the partial fulfillment for the academic requirement of

5th Semester B.E.
in

Software Design and Modeling Laboratory (18CSL58)

Submitted by:

Aniket Saha	2GI19CS189
Ashutosh Joshi	2GI19CS191
K. Rahul	2GI19CS190
Mohit Saboji	2GI19CS073

under the guidance of:
Prof. Arundhati Nelli
Department of Computer Science and Engineering,
KLS GIT,Belagavi

KARNATAK LAW SOCIETY'S
GOGTE INSTITUTE OF TECHNOLOGY
UDYAMBAG, BELGAUM-590008
(An Autonomous Institution under Visvesvaraya Technological University, Belgaum)
(APPROVED BY AICTE, NEW DELHI)

Department of Computer Science and Engineering

Certificate

This is to certify that the Course Project work titled **"Library Management System"** carried out by **Aniket Saha, Ashutosh Joshi, Mohit Saboji, K. Rahul** bearing USNs: **2GI19CS189, 2GI19CS191, 2GI19CS072, 2GI19CS190** is submitted in partial fulfilment of the requirements for 5th semester B.E. in **COMPUTER SCIENCE AND ENGINEERING**, as per the project requirement for the academic year 2021-22 under the guidance of **Prof. Arundhati Nelli, KLS GIT, Belagavi**. It is certified that all corrections/ suggestions indicated have been incorporated in the report. The course project report has been approved as it satisfies the academic requirements prescribed for the said degree.

Signature of Project Guide: _____

Date: _____

Prof. Arundhati Nelli,
Department of Computer Science and Engineering,
KLS Gogte Institute of Technology, Belagavi.

Name of the Examiners   Signature of the Examiners

1.   _____   1.   _____

2.   _____   2.   _____

TABLE OF CONTENTS

PROBLEM STATEMENT

To develop an application using Java to accomplish all library functions which were originally performed by the librarians. The application also has an associated relational database that stores all relevant information about the actors of the system and the books of the library.

THEORY

Library computerization is the process of adequately utilizing computer hardware cum software's and its varying facilities such as networking cables, software packages etc. in accomplishing library functions which were originally performed manually by the librarians.

Now we are in the age of automation. In this age we feel the importance of automation in all steps of daily jobs. We are trying to develop an automation system which will provide lots offacilities. The total automation system divided into many modules. Here our part is "Computerization of library activities and book transactions". This is a small part of total automation system but computerization of library activities will provide an environment which facilitates the user, easy to access the library information.

Computerization of libraries has provided libraries with cheaper means of purchasing information resources for their libraries. This is so as other methods where the libraries have toapply the use of manual means in selecting and acquiring its resources is more costly and involves more human efforts. Thus, libraries now acquire information resources through the World Wide Web and Internet, this has made it more affordable for libraries to procure more current information resources such as journals, magazines which are very costly when procuredin physical format. Computerizing the library functions and services encompasses the initial:

- Planning processes

- Selection of hardware and software

- Acquisition of selected hardware and software

- Training of library personnel on the use

- Maintenance of these novel system

- Networking of computer terminals (inter linking of computers)

- Retrospective conversion of the library collections i.e., printed materials and artifacts records by means of computer software into machine readable options.

PURPOSE

The library management system is a software to manage manual functions of a library. The application provides a full-fledged software that implements all functionalities of a library. The system aims to provide an environment to maintain the details of books and library members i.e., students and the librarian. The software helps to manage the entire library operations from maintaining book records to issue a book. In addition, it allows streamlined management of fine details of books such as author name, edition, and many other important details. So, it is easier to search for books and find the right materials for students and the librarian. The librarian performs tasks such as adding books and managing the users. The users perform activities such as issuing a book, searching for a book, extending the date of return, and returning a book.

The electronic management via the software is essential to track information like issue date, due date, who has borrowed any material, etc. The system is developed and designed with an aim to facilitate efficient management to the schools to manage a modern library with accurate data management.

ADVANTAGES

The library management system offers ease to perform day to day library operations electronically. this practice being many advantages like;

- Simple and easy to operate
- Mobile access, anytime, anywhere
- Search, add, update, and view library materials online
- Helps to manage library functions constructively
- Saves time and reduces overheads
- Reduce library's operating cost
- Remove manual processes to issue books and maintain records

DISADVANTAGES

The advantage of library management software outweigh the disadvantages. But there are few disadvantages associated with the system.

- Online stored data is prone to cyber hacks.
- Complications
- Risk of computer virus
- Since the system stocks data on the computer's hard drive, this raises the risk of data loss.

PERFORMANCE REQUIREMENTS

SPEED AND LATENCY REQUIREMENTS

- Data would be inserted in MySQL database within few seconds
- Query would respond on time and bring the results within few seconds
- UI should load within no time

PRECISION AND ACCURACY REQUIREMENTS

- After successful login, legitimate details have to be shown
- All the data should store accurate and valid information to the database
- Only registered users can update their information

CAPACITY REQUIREMENTS

- There is no user limit

DEPENDABILITY REQUIREMENTS

RELIABILITY REQUIREMENTS

- The user should register as a new user and update database with given information
- Only accurate information can give access of the system
- Every user can update their own information
- The system should perform activities immediately upon user response

MAINTAINABILITY AND SUPPORTABILITY REQUIREMENTS

MAINTENANCE REQUIREMENTS

- Modify the system when the application environment changes
- Fix bugs within the system
- Fix accidental data mistakes by the user

SUPPORTABILITY REQUIREMENTS

- Provide user guidance
- For better user experience, provide customer support

SECURITY REQUIREMENTS

ACCESS REQUIREMENTS

- Only registered users can login to the system
- Only the librarian can manage the user and book details

USABILITY REQUIREMENTS

EASE OF USE REQUIREMENTS
- System user interface should be extremely user friendly
- New user should not face much difficulties
- Profile maintenance and documentation support is not complex

UNDERSTANDABILITY AND POLITENESS REQUIREMENTS
- All kinds of users should understand the system with great ease
- Data of users is kept confidential and is highly prioritized

ACCESSIBILITY REQUIREMENTS
- The system should be accessible from all devices.
- There is minimal downtime
- The system responds quickly to user actions with less delay

FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS

A **Functional Requirement** (FR) is a description of the service that the software must offer. It describes a software system or its component. They describe the functionality of a system – how a system should react to particular set of inputs and what should be the corresponding output.

1. Requirement 1
   - Description: Login
   - Input: Credentials such as username and password
   - Output: Login successful

2. Requirement 2
   - Description: Logout
   - Input: Click event on logout button
   - Output: Logout successful

3. Requirement 3
   - Description: Adding new user
   - Input: User details
   - Output: Registration successful

4. Requirement 4
   - Description: Search for a book
   - Input: Book name
   - Output: Search results, either found or not found

5. Requirement 5
   - Description: Issue a book
   - Input: Book details and user account details
   - Output: Appropriate message, either success or failure

6. Requirement 6
   - Description: Add a new book (only for the admin)
   - Input: Book details
   - Output: New book added successfully

7. Requirement 7
   - Description: Search for a book
   - Input: Book name
   - Output: Search results, either found or not found

8. Requirement 8
    - Description: Return a book
    - Input: Book details and user account details
    - Output: Return successful

NON- FUNCTIONAL REQUIREMENTS:

Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

1. Availability
   - The system must be available 24 hours a day
   - Database backup and recovery plan should be proper in order to avoid anyunexpected downtime of application.

2. Security requirements
   - User authentication and validation of members using their unique member ID
   - Proper accountability which includes not allowing a member to see othermember's account
   - System will be connected to secured database

3. Performance requirement
   - The information is refreshed depending upon whether some updates haveoccurred or not in the application
   - The system shall respond to the member a short time of the request submittal
   - The system shall be allowed to take more time when doing large processing jobs.
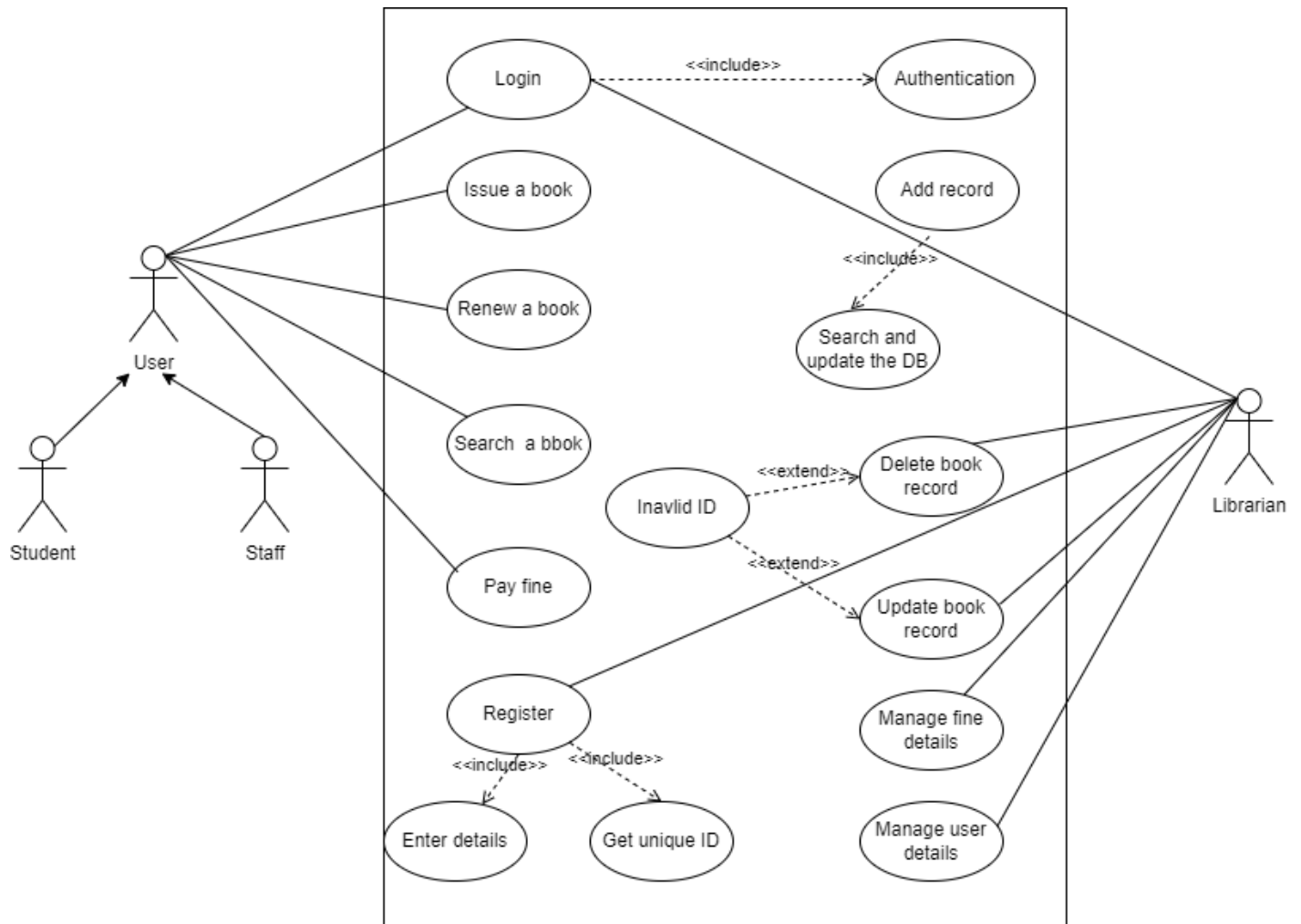
4. Hardware Requirements
   - Processor: Minimum 1 GHz or more
   - Minimum 256MB of RAM
   - Minimum 1GB of free hard drive space

5. Software Requirements
   - Java SE 11 or higher
   - Apache NetBeans IDE 12.5
   - MySQL server
   - Operating System: Windows XP and higher

USE CASE DIAGRAM

USE CASE DESCRIPTION

REGISTRATION

Use Case Name	Registration	
Scenario	User registration to the system	
Brief Description	User should register providing all details with username and password	
Actors	Student, Staff, Librarian	
Pre-condition	All details are valid	
Post-condition	User details added to the database	
Flow of events	User	System
	• Must provide all details • Username and password are valid • All required fields are filled	• Confirm valid username • Confirm unique username
Exception Condition	Username already exists Credentials are not valid	

LOGIN

Use Case Name	Login	
Scenario	User login to the system	
Brief Description	User should login with valid username and password	
Actors	Student, Staff, Librarian	
Pre-condition	The user with the username already exists or is registered	
Post-condition	After successful login, the user has access to the system	
Flow of events	User	System
	• Username and password are valid	• Confirm valid username • Confirm unique username • Give access of the system to the user
Exception Condition	Username does not exist Username and/ or password do not match	

MANAGE USER DETAILS

Use Case Name	Manage user details	
Scenario	Librarian can check borrowed books and the corresponding details	
Brief Description	Librarian can access user account details that has borrowed a book and can impose fine if applicable	
Actors	Librarian	
Pre-condition	Librarian is already registered	
Post-condition	The books that are currently borrowed, and their details is visible to the librarian	
Flow of events	Librarian	System
	• Login to the system • Check currently borrowed books • Check corresponding user details • Update fine details	• Authenticate librarian • Give access of the system • Show books that have been borrowed • Show corresponding user details
Exception Condition	Username does not exist Username and/ or password do not match	

ADD RECORD

Use Case Name	Add record	
Scenario	Librarian adds new books	
Brief Description	Librarian can add new books into the database and its details	
Actors	Librarian	
Pre-condition	The book with the same name does not already exist in the database	
Post-condition	The new book is successfully added into the database	
Flow of events	Librarian	System
	- Login to the system - Add a new book - Enter book details	- Authenticate librarian - Give access of the system to the user - Add a record of the newly added book to the database
Exception Condition	Username does not exist Book with the same name and details already exist in the database	

SEARCH FOR A BOOK

Use Case Name	Search for a book	
Scenario	User (student, staff) can search a book by its name, author name, edition	
Brief Description	A user can search for a book by providing appropriate details such as book name or author name. If the book exists, the book information is displayed	
Actors	User (student, Staff)	
Pre-condition	The book with the same name already exists in the database	
Post-condition	The book information is displayed to the user	
Flow of events	User	System
	• Login to the system • Search for a book • Select keyword • Enter book details	• Authenticate user • Give access of the system to the user • Search for the book by the specified keyword in the database • Display book information
Exception Condition	Book does not exist	

ISSUE A BOOK

Use Case Name	Issue a book	
Scenario	User (student, staff) can issue a book	
Brief Description	User can issue a book for himself and may choose to return it or renew it on or before the specified date from the issue data	
Actors	User (student, Staff)	
Pre-condition	The book exists in the database and sufficient copies of the book are available for issue	
Post-condition	The book is issued to the user, user account details and book details get updated and date of return is computed	
Flow of events	User	System
	• Login to the system • Search for a book • Request for issue	• Authenticate user • Give access of the system to the user • Search for the book in the database • Check if book can be issued • Update user, book details • Compute return date
Exception Condition	Book does not exist No copies of the book are available for issue	

RETURN A BOOK

Use Case Name	Return a book	
Scenario	User (student, staff) can return a book already issued to them	
Brief Description	User can return a book already issued to them. If the user has crossed the date of return, then the user is subjected to additional fine	
Actors	User (student, Staff)	
Pre-condition	The user has borrowed the book	
Post-condition	The date of return is checked for the computed date, and if there is any difference the user is prompted to pay the fine	
Flow of events	User	System
	• Login to the system • Request for return • Select books among the borrowed books • Pay fine if prompted	• Authenticate user • Give access of the system to the user • Check for an entry in the borrowed books table • Compute fine details • Update user, book details
Exception Condition	User not valid Selected book not borrowed	

RENEW A BOOK

| Use Case Name	Renew a book
Scenario	User (student, staff) can renew a book
Brief Description	User can choose to renew a book and extend the date of return only for one time.
Actors	User (student, Staff)
Pre-condition	The user has borrowed the book and has not applied for renewal earlier
Post-condition	The date of return is extended and changes are made in the tables
Flow of events	<table><tr><td>User</td><td>System</td></tr><tr><td> - Login to the system - Request for renewal - Select book </td><td> - Authenticate user - Give access of the system to the user - Check if user has borrowed the book - Check for earlier renewal requests - Update user, book details - Compute the new return date </td></tr></table>
Exception Condition	User not valid The selected book is not borrowed The selected book has already been renewed

UPDATE BOOK DETAILS

| Use Case Name	Update book details	
Scenario	Librarian can update book details	
Brief Description	Librarian can update book details such as name, author name, edition, number of copies, etc.	
Actors	Librarian	
Pre-condition	The book already exists in the database	
Post-condition	Appropriate changes are made in the books table	
Flow of events	Librarian	System
	- Login to the system - Search for a book - Update details	- Authenticate user - Give access of the system to the user - Search for the book in the database - Update details
Exception Condition	User not valid Book does not exist	

DELETE A BOOK

| Use Case Name	Delete a book	
Scenario	Librarian can delete a book from the books table of the database	
Brief Description	Librarian can delete a book if it no longer needs to be kept in the library for obvious reasons	
Actors	Librarian	
Pre-condition	The book already exists in the database	
Post-condition	The selected book is removed from the books table of the database	
Flow of events	Librarian	System
	• Login to the system • Search for a book • Delete book	• Authenticate user • Give access of the system to the user • Search for the book in the database • Remove the record of the selected book • Update book table
Exception Condition	User not valid Book does not exist	

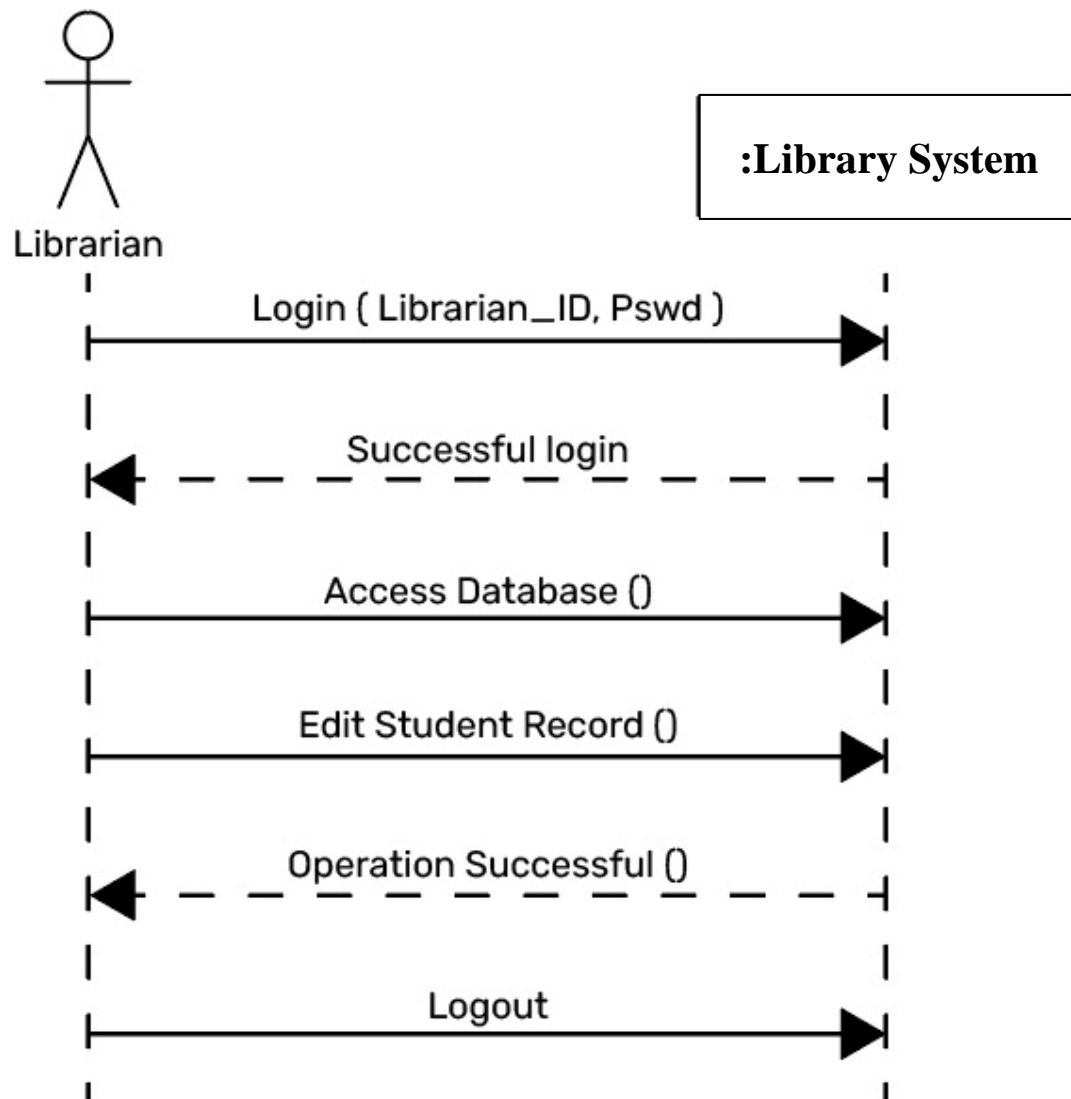# SEQUENCE DIAGRAM
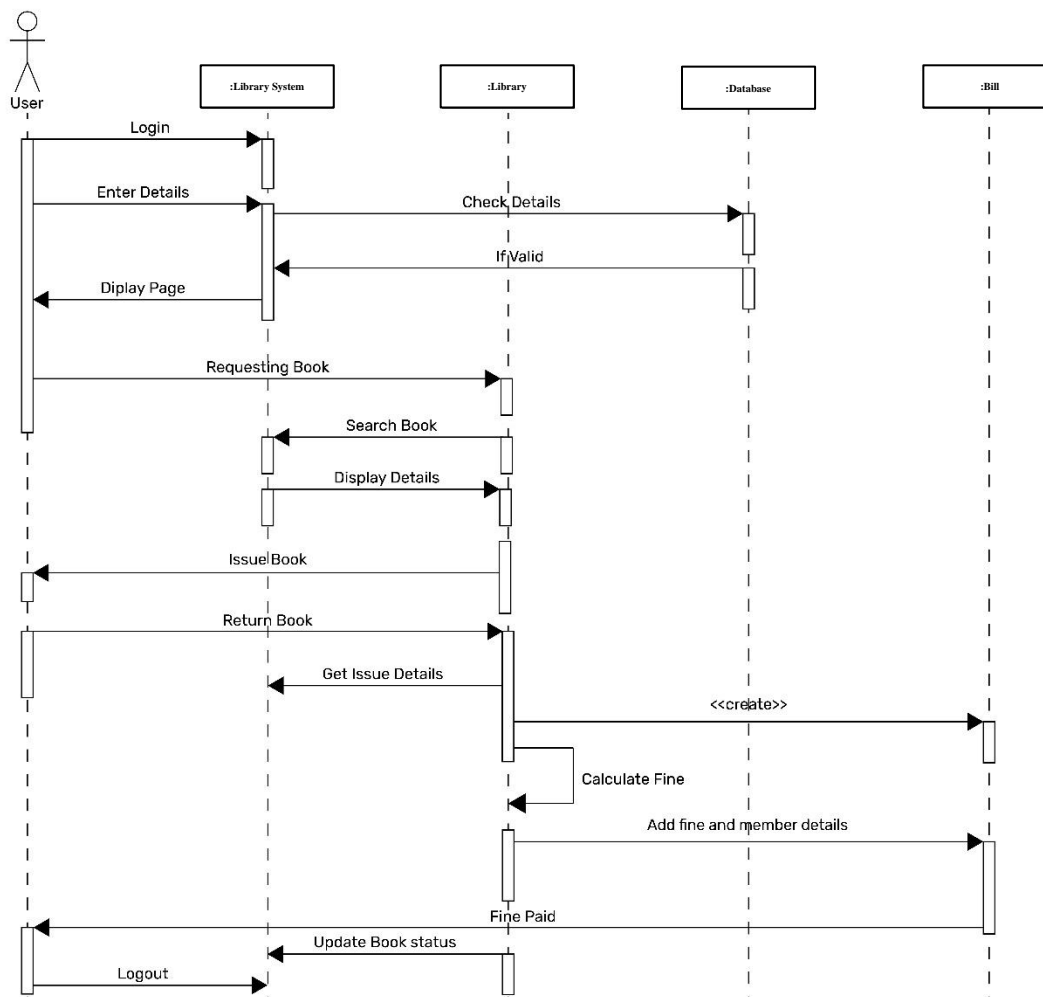
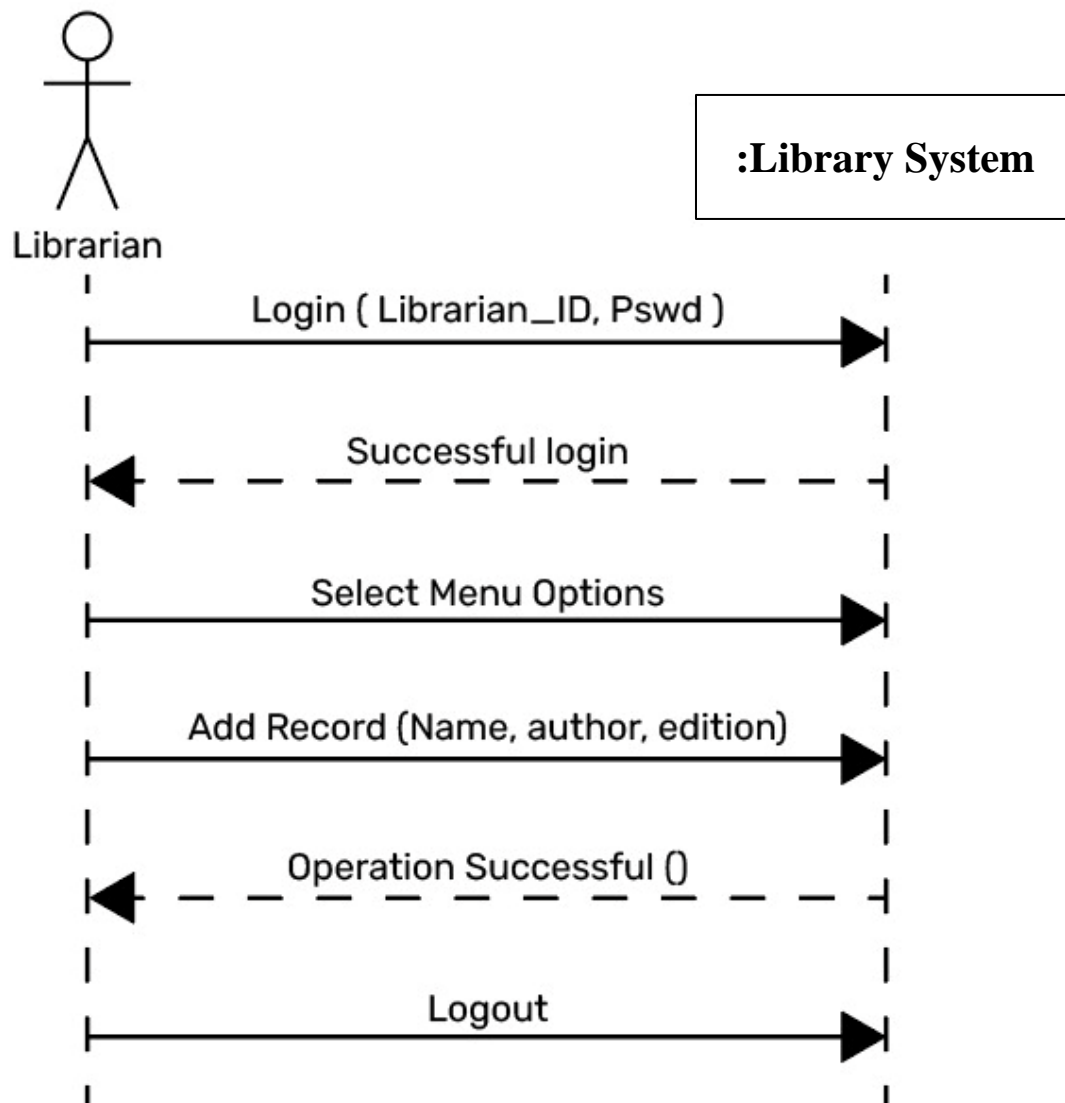## VIEW/EDIT BOOK DETAILS
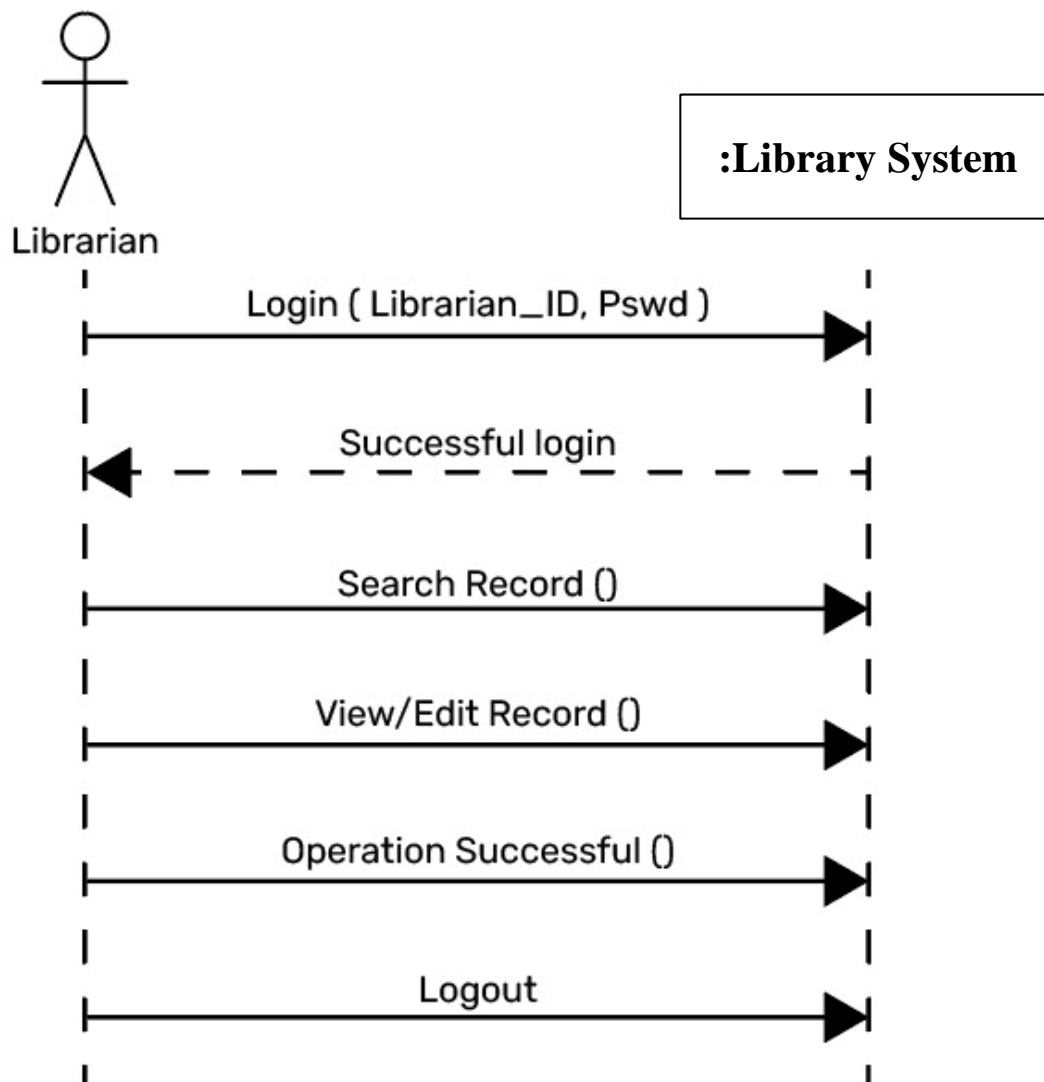
View/Edit Book Detail Sequence Diagram

ADD MEMBER
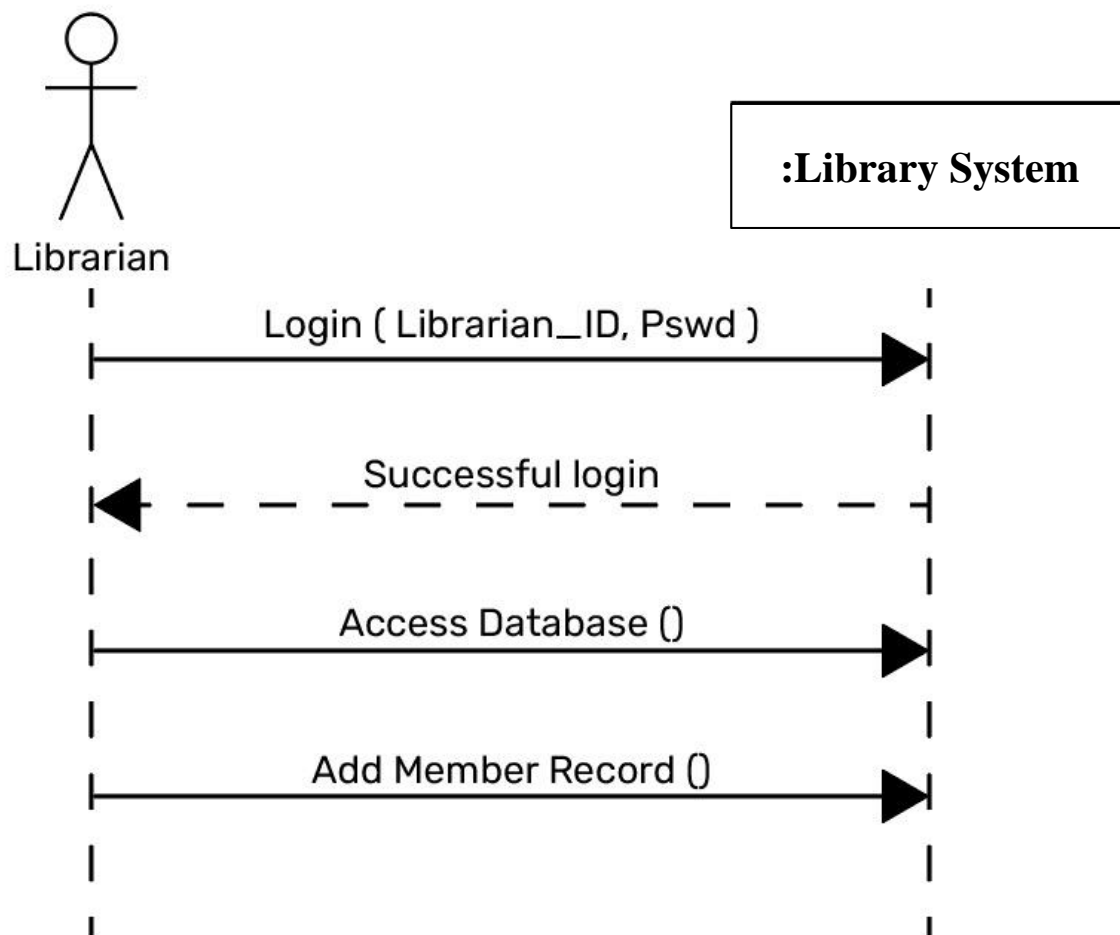
Add Member Sequence Diagram

EDIT STUDENT RECORD

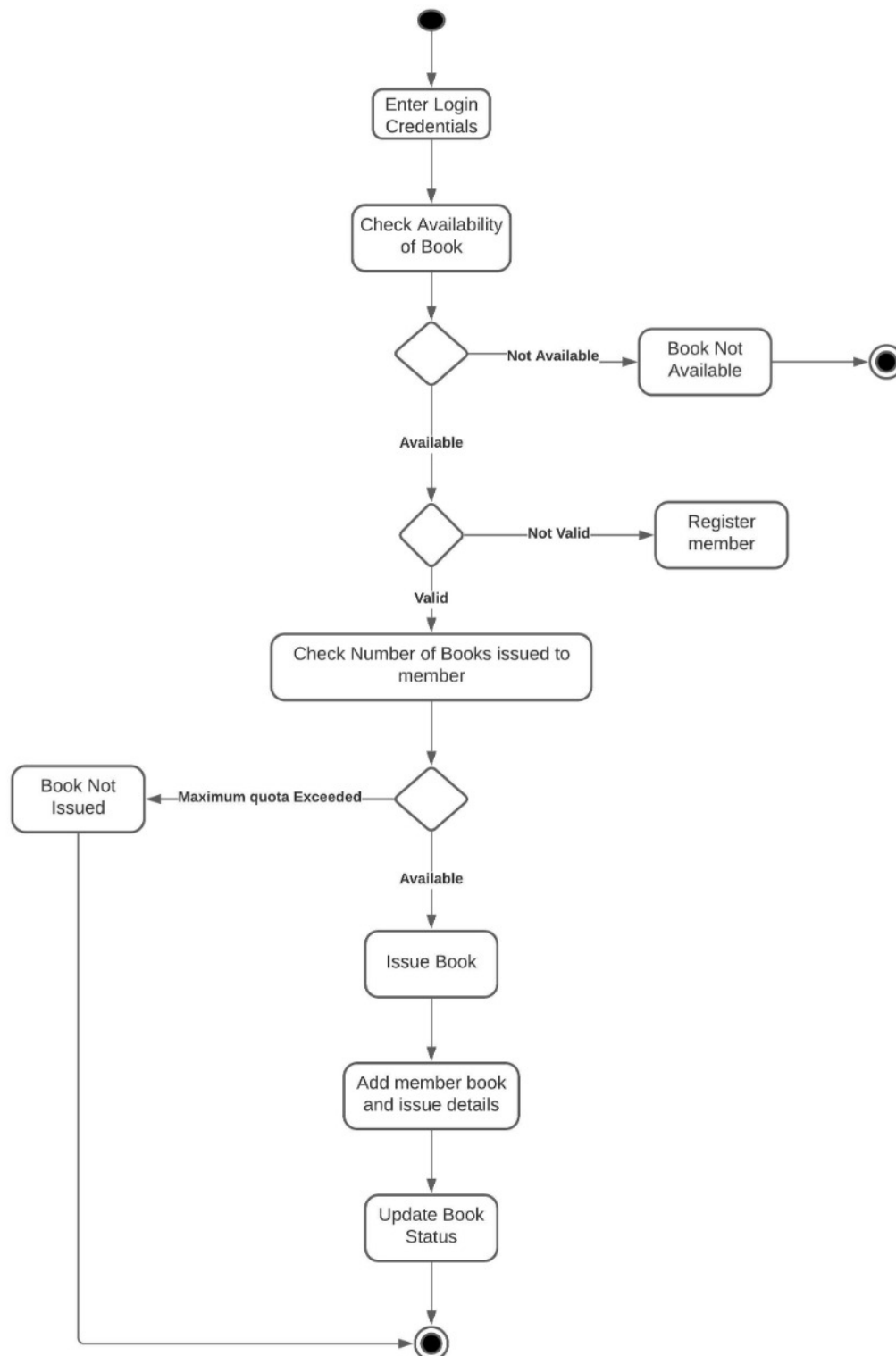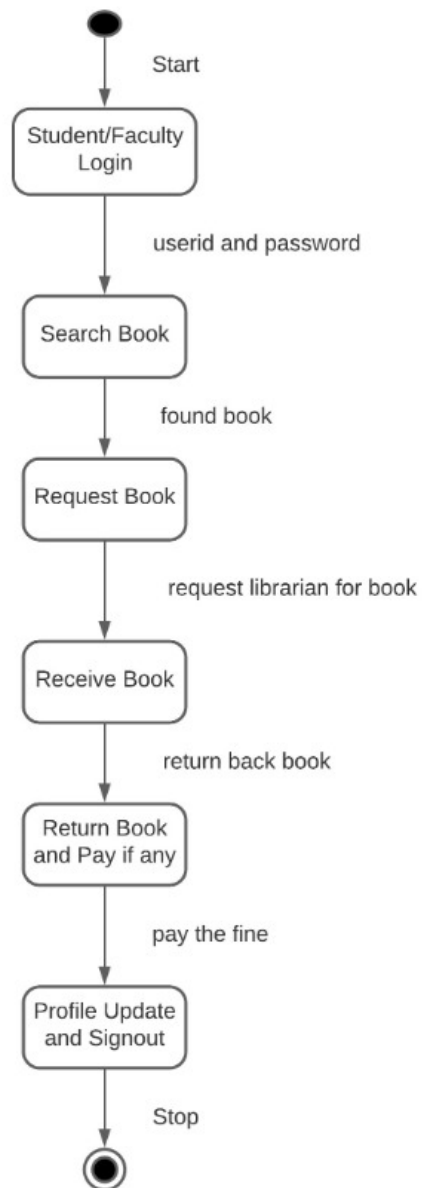Edit Student Record Sequence Diagram

ADD BOOK RECORD

Librarian

:Library System

Login (Librarian_ID, Pswd)

Successful login

Select Menu Options

Add Record (Name, author, edition)
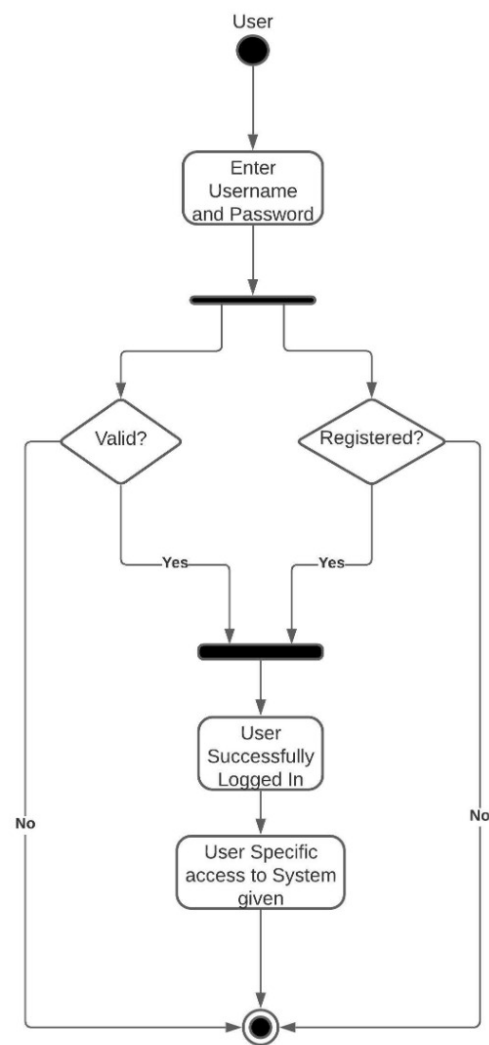
Operation Successful ()

Logout

Add Book Record Sequence Diagram

ACTIVITY DIAGRAM

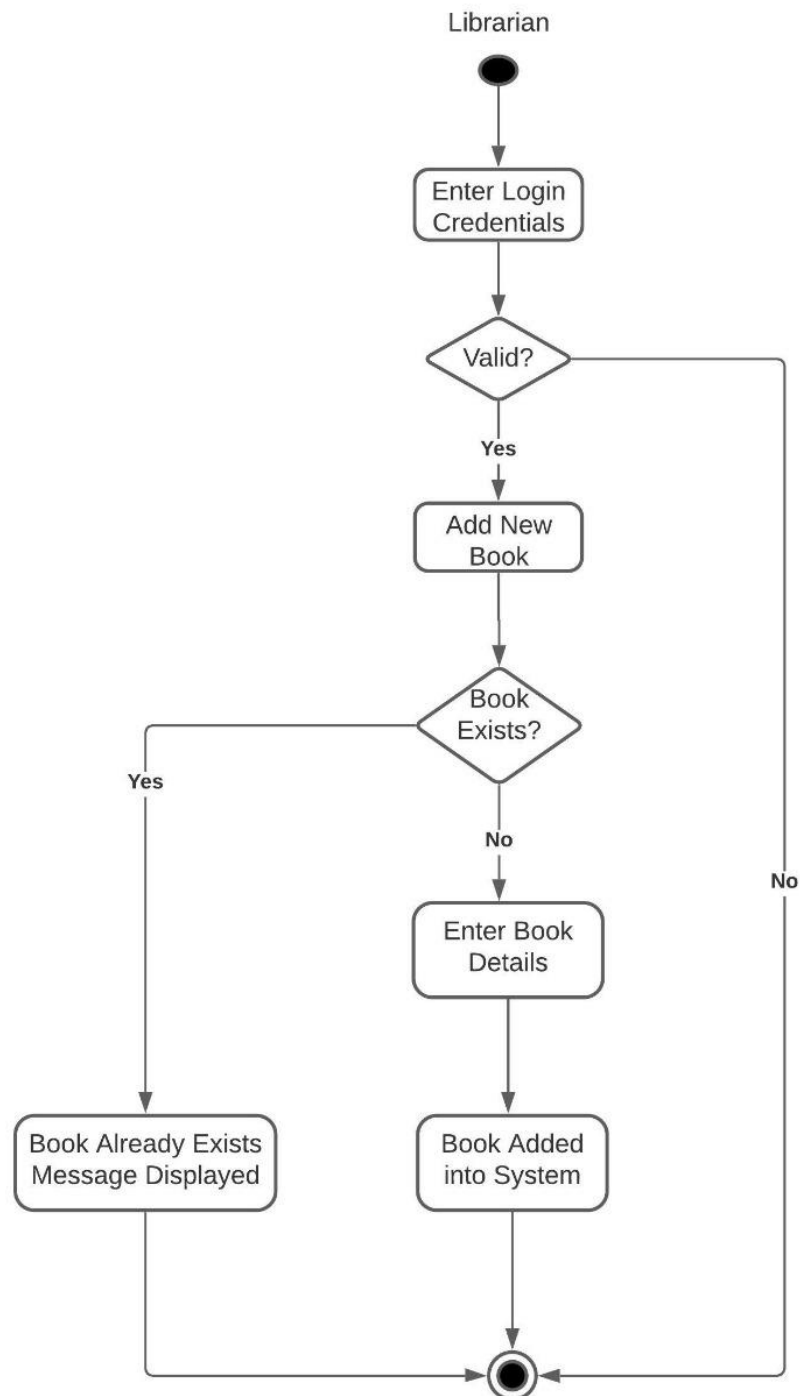# ISSUE/ RETURN A BOOK
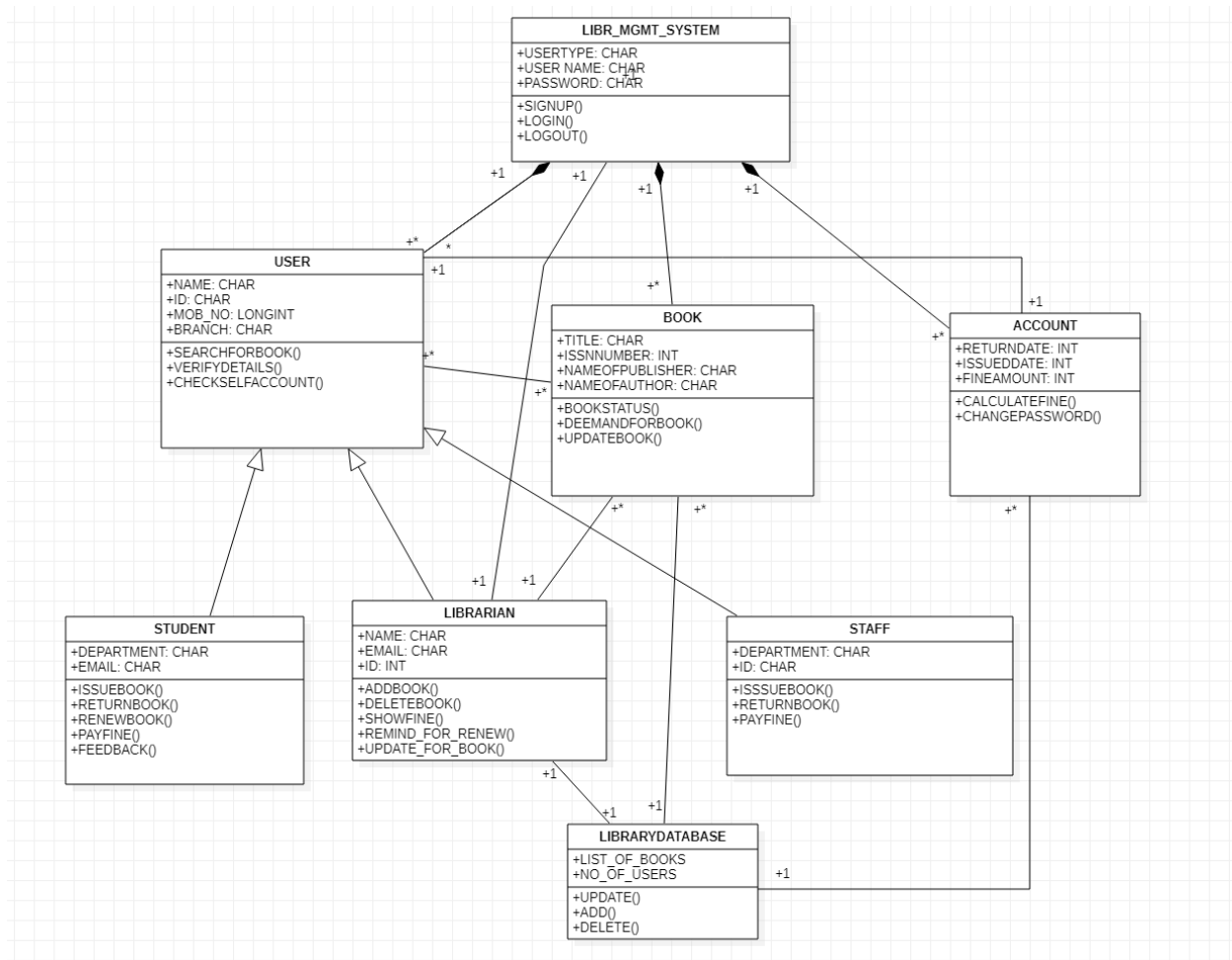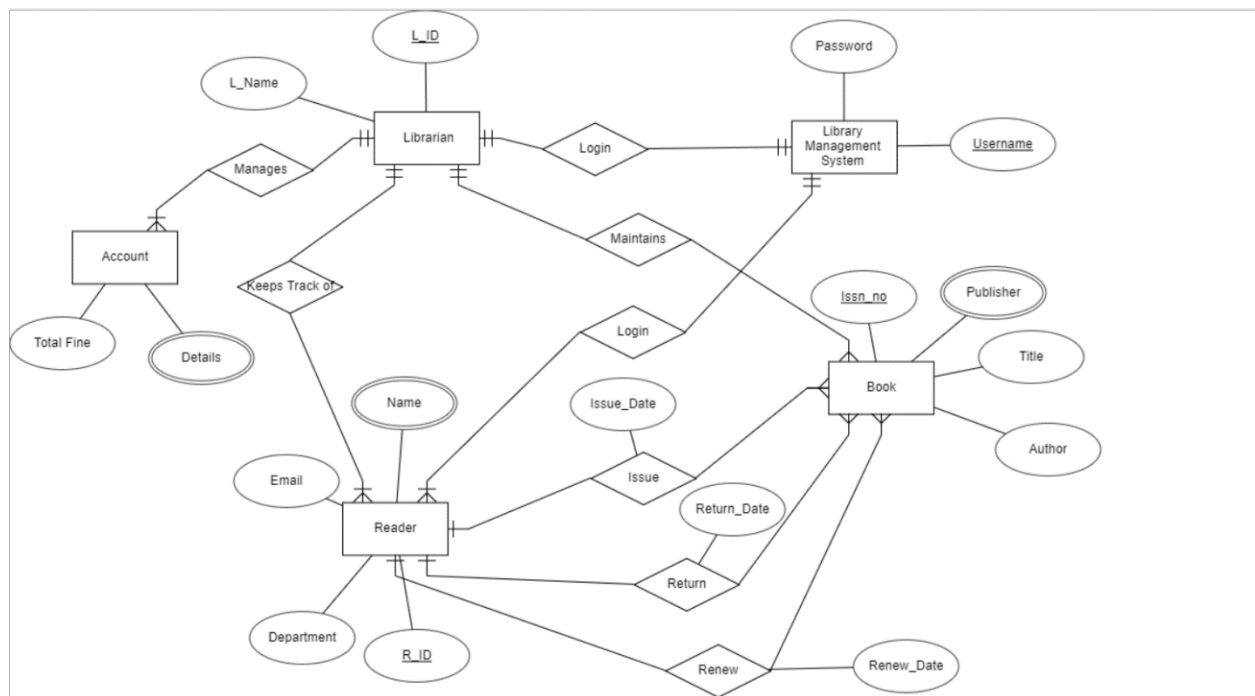
LOGIN

SIGN UP

ADD A BOOK

ADD A BOOK

CLASS DIAGRAM

LIBR_MGMT_SYSTEM
+USERTYPE: CHAR
+USER NAME: CHAR
+PASSWORD: CHAR

+SIGNUP()
+LOGIN()
+LOGOUT()

USER
+NAME: CHAR
+ID: CHAR
+MOB_NO: LONGINT
+BRANCH: CHAR

+SEARCHFORBOOK()
+VERIFYDETAILS()
+CHECKSELFACCOUNT()

BOOK
+TITLE: CHAR
+ISSNNUMBER: INT
+NAMEOFPUBLISHER: CHAR
+NAMEOFAUTHOR: CHAR

+BOOKSTATUS()
+DEEMANDFORBOOK()
+UPDATEBOOK()

ACCOUNT
+RETURNDATE: INT
+ISSUEDDATE: INT
+FINEAMOUNT: INT

+CALCULATEFINE()
+CHANGEPASSWORD()

STUDENT
+DEPARTMENT: CHAR
+EMAIL: CHAR

+ISSUEBOOK()
+RETURNBOOK()
+RENEWBOOK()
+PAYFINE()
+FEEDBACK()

LIBRARIAN
+NAME: CHAR
+EMAIL: CHAR
+ID: INT

+ADDBOOK()
+DELETEBOOK()
+SHOWFINE()
+REMIND_FOR_RENEW()
+UPDATE_FOR_BOOK()

STAFF
+DEPARTMENT: CHAR
+ID: CHAR

+ISSSUEBOOK()
+RETURNBOOK()
+PAYFINE()

LIBRARYDATABASE
+LIST_OF_BOOKS
+NO_OF_USERS

+UPDATE()
+ADD()
+DELETE()

ER DIAGRAM

SCREENSHOTS