

Practical 1.

Aim:- Select and write down the problem Statement for library management System.

Problem Statement:-

Manual process of keeping student records, book records, account details, managing employee is very difficult. There are various problems also faced by the student in library such as finding any particular book, ~~using~~ information whether book is available or not, for what time this book will be available, searching of books using ISBN number etc. To eliminate this manual system, library management system has been developed. It will handle all the current issues faced by the students and by its admin personnel.

Purpose

To store all the information in the database from where user will place their query and get the results on the basis of their query. Only valid users will be able to access this system. Through this system it will be easy to manage accounts and various details of particular student and employees working under library along with the records of book.

Overview

This system will have login page from where its user can access. This page will provide login for admin, working staff members and the students. Staff members accounts will be managed by the admin. To access the library resources students have to register by using their registration number, email address, phone number, class roll no. & password. After successful registration they will be provided the login facility.

Students can search books by using book ISBN no. or by author name or by title of book along with author names. After completion of this process students will be provided with book details such as where it is located by using location no. and by their row and column numbers.

Admin will be able to add staff, delete staff, add students, delete students, add books, manage account details, schedule working time table etc.

Practical 2.

Aim:- Analyze requirement for a system and develop Software Requirement Specification Sheet (SRS) for suggested System.

Software Requirement Specification (SRS) for Library Management System.

1.0 Introduction

1.0.1 Purpose :-

The main objective of this document is to illustrate the requirements of the project library management system. The document gives the detailed description of the both functional and non-functional requirements proposed by the client.

The purpose of this project is to provide a friendly environment to maintain the details of books and library members also this project maintains easy circulation system using computers and to provide different reports. It describes the hardware and software interface requirements using UML models and UML.

1.0.2 Scope of the Project :-

Library management system is basically updating the manual library system into an internet-based web application so that the user can know the details

of their accounts, availability of books and maximum limit for borrowing and many more features.

The project is specifically designed for the use of librarians and library users. The product will work as a complete user interface for library management process and library usage from ordinary user. It can be used by any existing or new library to manage its books and book borrowing, insertion and monitoring. It is especially useful for any educational institute where modifications in the content can be done easily according to requirements.

1.3 References:-

Books:-

- Software Requirements (Microsoft) Second edition by Karl G. Wiegers.
- Fundamentals of Database System By Elmasri

2. Description:-

2.1 Product Perspective:-

LMS is a replacement for the ordinary library management systems which depend on paper work for recording book and user's information. LMS will provide an advanced book search mechanism and will it easy to borrow, insert and index a book in the library.

Teacher's Signature :

2.2 Product Functions:

The proposed library Management System (LMS) is designed to simplify the day-to-day activities of a library, providing features for both users and administrators.

• Authentication and Authorization System:

The system implements a secure login mechanism for users, and administrators. The admin has the authority to manage user access and ensure data integrity.

for users:

- (i) New User Registration: This feature allows new users (Students, teachers, etc.) to sign up for the system by providing the necessary details.
- (ii) Student Login: This feature provides authenticated access for registered users to use the system.
- (iii) Search Book: This feature allows the user to search for books based on criteria such as Book ID, Book name, or author name, enhancing the ease of locating desired materials.
- (iv) Issue Book: This feature allows users in borrowing books from the library by recording the transaction and updating the availability status.

(V) Return Book: This feature allows users to return books. It records various details into the system, such as book ID.

→ for admin:-

- Record library Activities.
- Manage books.
- Manage Student.
- view Issued Books.
- Defaulters list.
- Issue Books.

3. Assumptions and Dependencies:-

- The assumptions are:-
- The coding should be error free.
- The system should be user-friendly so that it is easy to use for the users.
- The information of all users, books and libraries must be stored in a database that is accessible by the website.
- The library system is running 24 hours a day.

→ Dependencies are:-

- The specific hardware and software due to which the product will be run.

- on the basis of listing requirements and specification the project will be developed and run.
- The end users (admin) should have proper understanding of the product.
- The system should have the general report stored.

4. Functional Requirements

- The LMS should store all information about Librarian and other users (Students and faculty members) like their login info, books issued etc.
- LMS should store all information about the books and users in two separated databases.

4.1 Software Requirements:

This software package is developed using HTML, bootstrap for front end. PHP and MySQL Server as the back end to store the database for back end we are using Xampp server.

- operating System:- windows 7, 8, 10, 11.
- language: HTML, CSS, JavaScript, PHP, SQL
- Database: MySQL Server (back end)

4.2 Hardware Requirements:-

- Processor:- Intel core i3 or above for a stable experience and fast retrieval of data.
- Hard disk:- 40 GB and above.
- Ram:- 2GB or more, recommended 2GB for fast reading and writing capabilities which will result in better performance times.

5. Non-functional Requirements.5.1 Usability Requirements:-

- Our user interface should be interaction simple and easy to understand. The system should prompt for the user and administrator to login to the applications for proper input criteria.
- library management system shall handle expected and non-expected errors in ways that prevent loss in information and long downtime periods.

5.2 Security Requirements:-

- System should use Secured Database.
- Normal users can just read information but they cannot edit or modify.

- Proper user authentication should be provided.

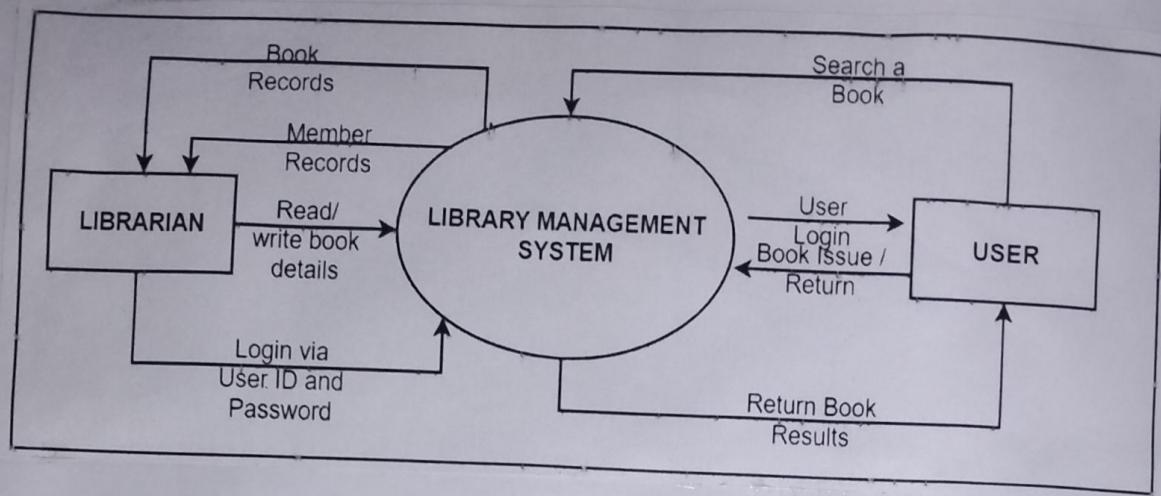
5.3 Performance Requirements:-

- The system shall accommodate high number of books and users without any fault.
- Responses to view information shall take no longer than 5 seconds to appear on the screen.

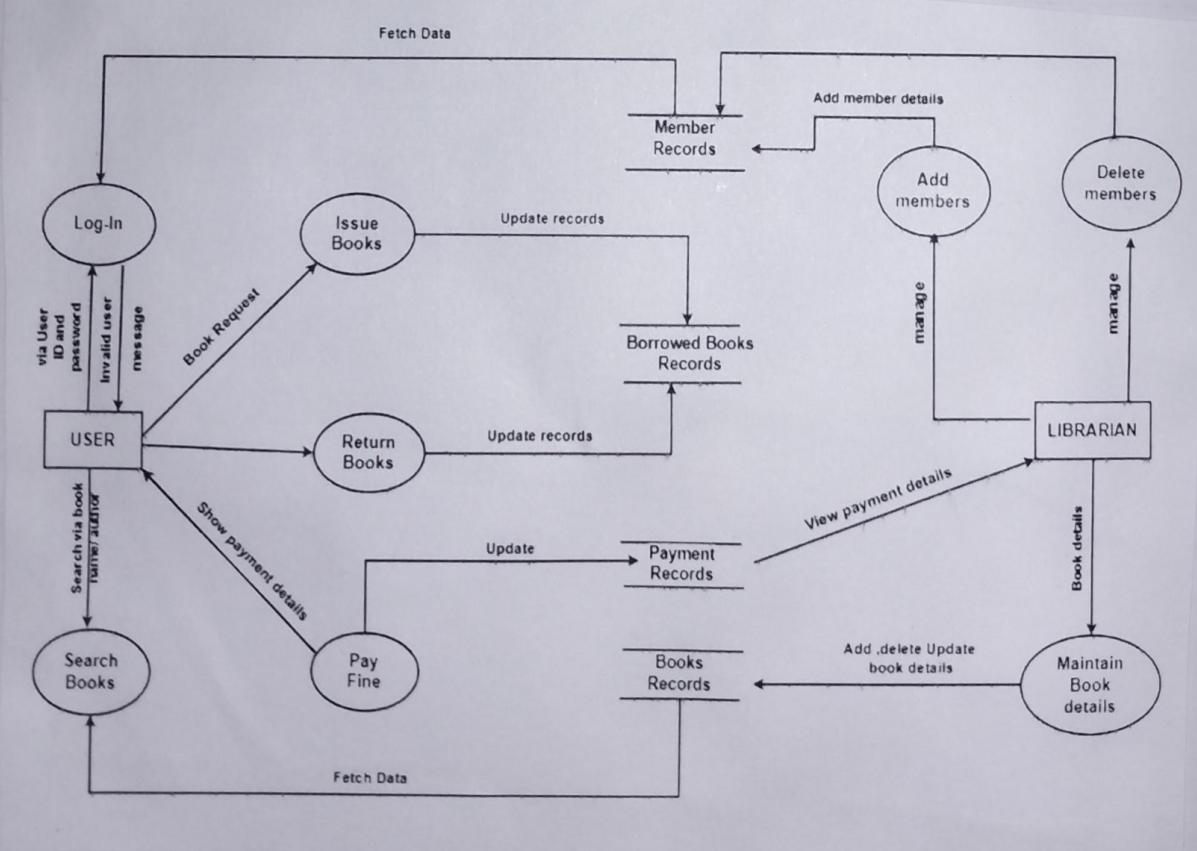
→ Appendix A :-

- A:- Admin, Abbreviation, Acronym, Assumptions.
- B:- Books, Business rules.
- C:- class, client, Conventions.
- D:- Data requirement, Dependencies.
- G:- GUI
- K:- Key. L:- Library, Librarian.
- N:- Non-functional Requirement.
- O:- operating environment
- P:- Performance, Perspective, Purpose;
- R:- Requirement, Requirement attributes.
- S:- Safety, scope, security, System features;
- U:- User, user class and characteristics, user requirements,

Level-0 DFD



Level-1 DFD



Practical-3

Aim:- To create the function oriented diagram: Data flow Diagram (DFD).

Theory:-

A Data flow Diagram (DFD) is a visual representation of the information flow through a process or system. DFDs help you better understand processes or system operations to discover potential problems, improve efficiency, and develop better processes.

The DFD of Library Management System.

- The inputs can be:-
- Book request when a student requests for a book.
- library card when the student has to show or submit his/her identity proof.

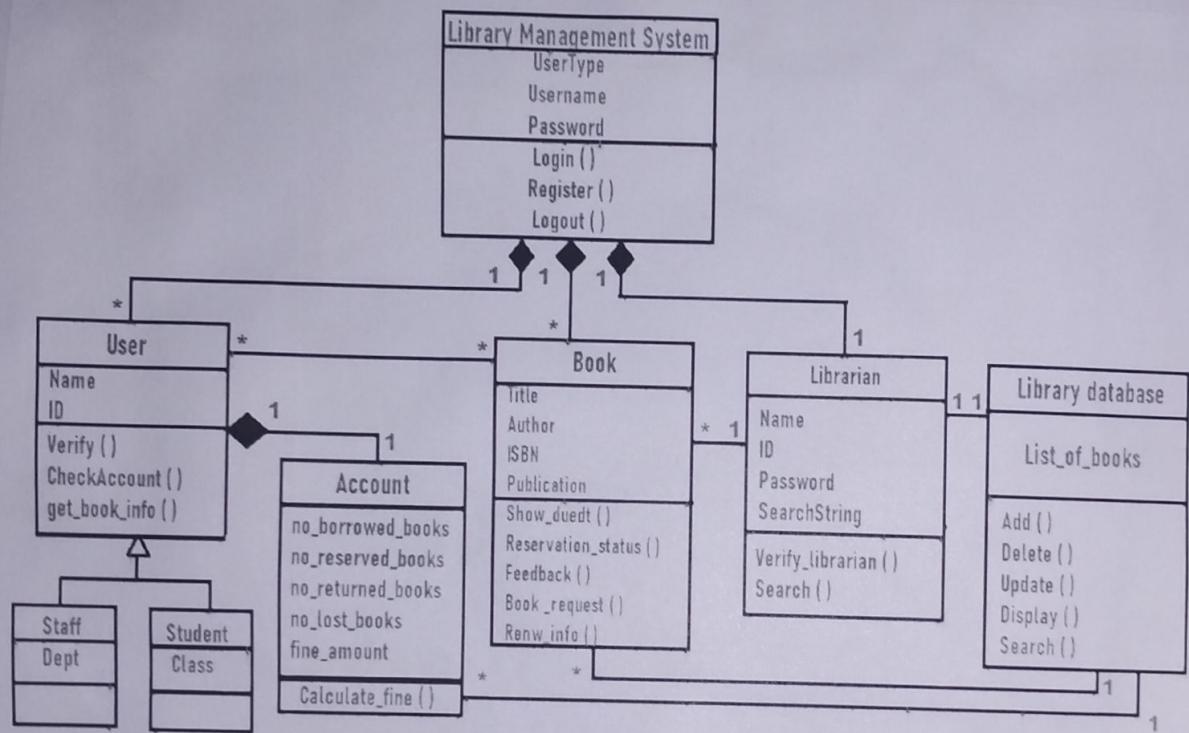
→ Components of level 0 DFD:

- External Entities:- Actors interacting with the system.
- System:- The main component of the system.
- Data Flow:- Relationships showing data movement.
- Processes:- High-level functions or use cases.

→ Components of level 1 DFD:

- Process:- Detailed functions or tasks performed by the system, representing lower-level functions composed to Level 0.

- Data Stores :- places where data is stored within the system.
- Data Flows :- Show the movement of data between processes, Data stores, and external entities
- External Entities :- Actors interacting with the System, similar to level 0.



CLASS DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM

Practical 4

Aim:- To Draw the Structural view diagram for the System: class diagram

Theory:-

Class diagrams are generally used for conceptual modeling of static view of a software application, and for modeling translating models into programming code in a detailed manner.

Class Diagram for Library Management System:

1) Aggregation:- Aggregation simply shows a relationship where one thing can exist independently of other thing. It means to create or compose different abstractions together, in defining a class. Aggregation is represented as a part of relationship in class diagram. In diagram given below, we can see that aggregation is represented by an edge with a diamond end pointing towards superclass "The Library management System". It is superclass that consists of various classes. These classes are user, Book, and Librarian as shown in diagram.

2) Multiplicity:- Multiplicity means that number of elements of a class is associated with another class. These relations can be one to one, many-to-many, and many-to-one or one-to-many.

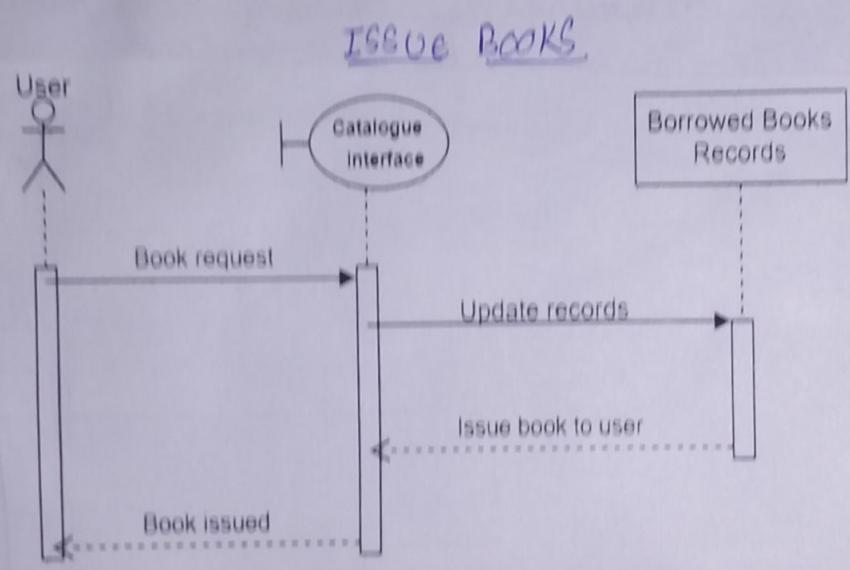
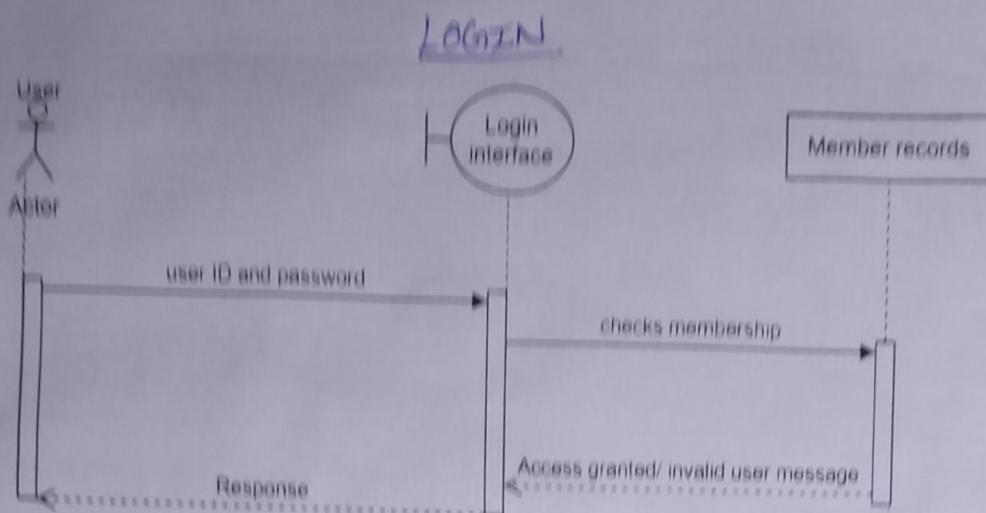
→ classes of Library Management System:-

- Library Management System class:- It manages all operations of library Management System. It is central part of organization for which software is being designed.
- User class:- It manages all operations of user.
- Librarian class:- It manages all operations of librarians.
- Book class:- It manages all operations of books. It is basic building block of System.
- Account class:- It manages all operations of account.
- library database class:- It manages all operations of account library database.
- Staff class:- It manages all operations of staff.
- Student class:- It manages all operations of student.

→ Attributes of library Management System:-

- library management System Attributes:- User type, username, password.
- User Attributes:- Name, Id.
- Librarian Attributes:- Name, Id, password, Search string.
- Book Attributes:- Title, Author, ISBN, Publication.
- Account Attributes:- no borrowed books, no issued books, no returned books, no lost books. Fine amount.
- library database Attributes:- List of books.
- Staff class Attributes:- Dept.
- Student class Attributes:- Class.

SEQUENCE DIAGRAMS OF LIBRARY MANAGEMENT SYSTEM



Practical - 5

Aim:- To perform the user's view analysis for
Suggested System: Sequence diagram.

Theory :-

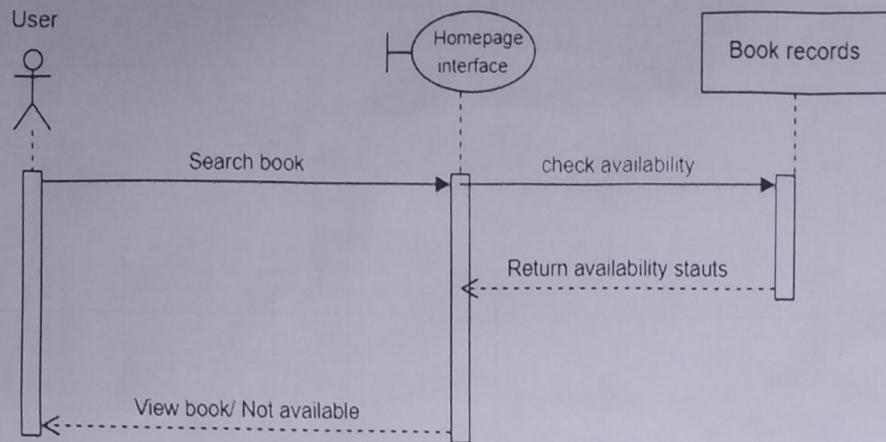
The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that those lifeline took part at the run time. The lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. It incorporates the iterations as well as branching.

→ Components of Sequence Diagram :-

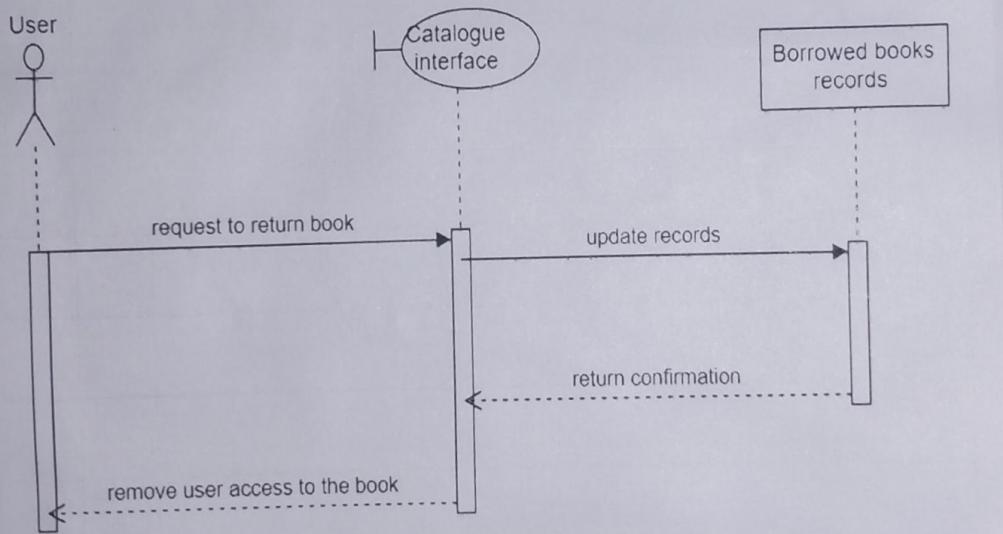
1.) Lifelines:- Representing Participants:

At the core of a sequence diagram are lifelines, which represent individual participants in the system. These participants can be objects, and their interactions are depicted over time. Lifelines are typically shown as rectangles containing the object's name. If the lifeline represents the classifier owning the sequence diagram, it may be labeled as "Self". Actors from use cases or elements from Robustness diagrams can also own lifeline

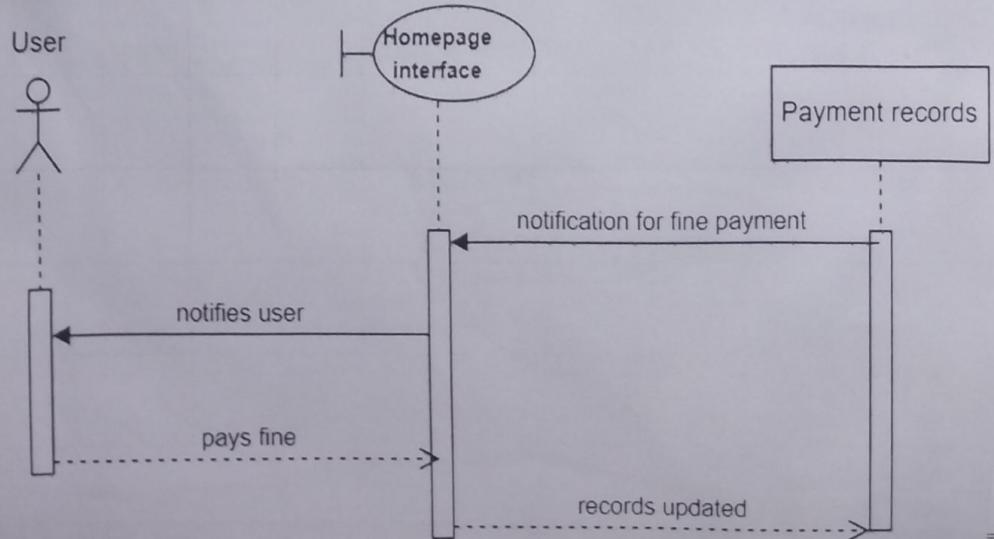
SEARCH Books



RETURN Books



PAY FINE



2. Messages: The Essence of communication

Messages are arrows indicating interactions between lifelines. They can be complete, lost, or found, synchronous or asynchronous, and can represent calls or signals. Understanding the nature of messages is crucial for deciphering the communication flow within a system.

3. Execution Occurrence: Activating Control Focus

A thin rectangle running down a lifeline denotes the execution occurrence or activation of a focus of control. This helps in visualizing when an object is actively involved in sending or receiving messages.

4. Self message: Recursive or Internal method calls

A self message is used to represent recursive calls of an operation or when one method within an object calls another method of the same object. It creates a nested focus of control within the lifeline's execution occurrence.

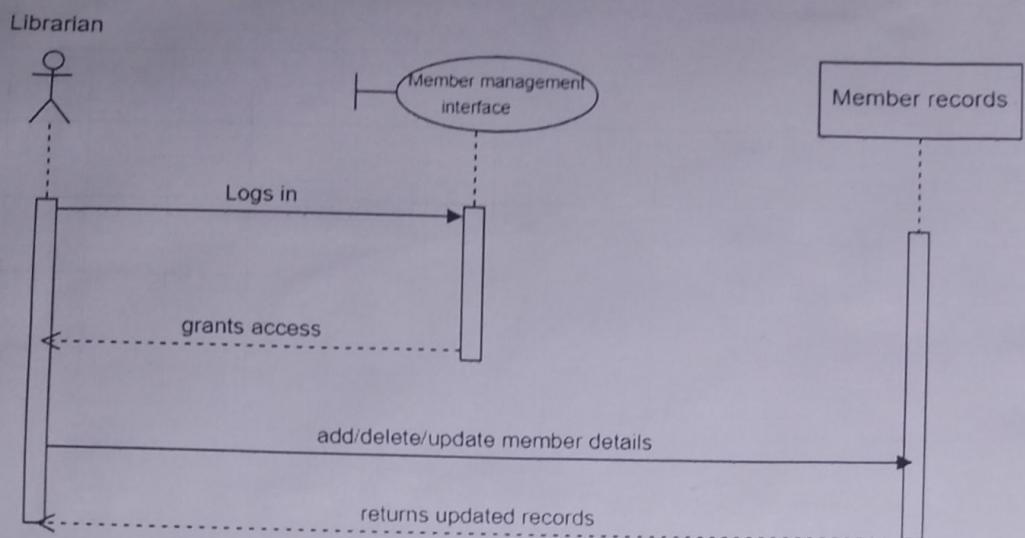
5.) Lost and found Messages: Navigating Uncharted territory

lost messages are those that do not reach their intended recipient, while found messages arrive from unknown senders. These are denoted by special symbols and indicate potential communication issues within the system.

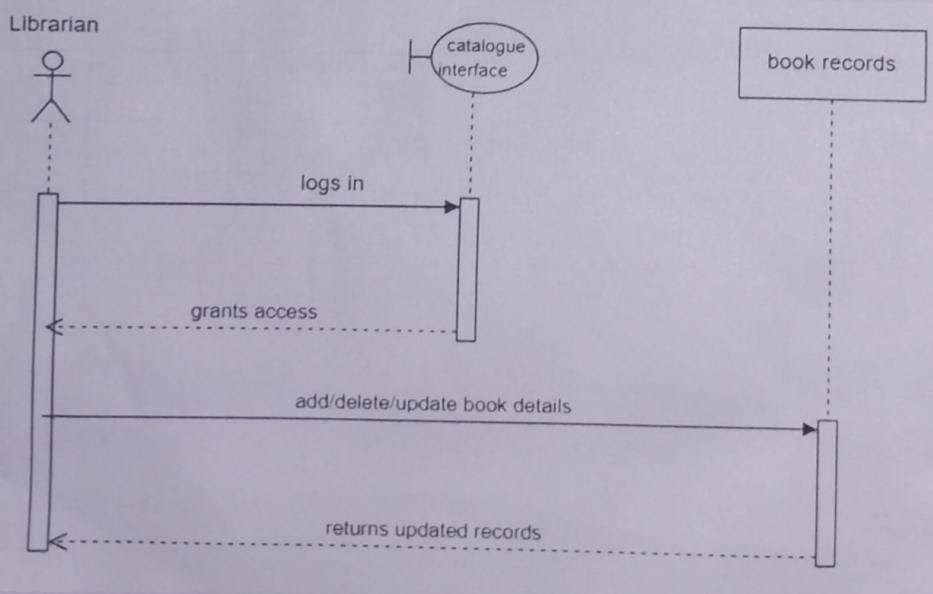
6.) Lifeline Start and End: The birth and death of lifelines

Lifelines can be created or destroyed during the timescale represented by a sequence diagram. The start and end of lifeline are marked by specific symbols, indicating creation or termination.

MEMBER MANAGEMENT



CATALOGUE MANAGEMENT



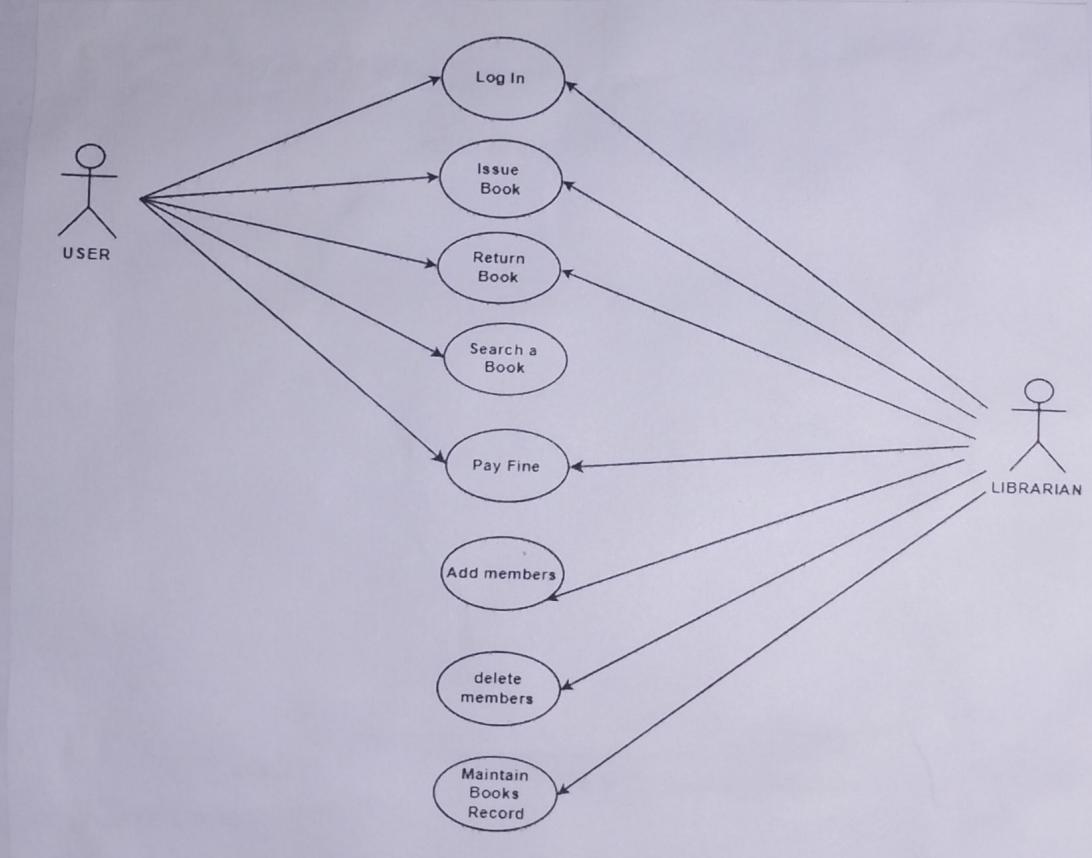
7.) Duration and Time constraints: Modelling Real-time Systems

To represent the passage of time in real-time systems or time-bound business processes, duration constraints can be applied to messages. This results in sloping lines to indicate the time taken for actions.

8.) Combined Fragments: Adding Procedural Logic:

While sequence diagrams are not intended for complex procedural logic, combined fragments offer a mechanism for introducing a degree of logic. These include alternative option, break, parallel, weak sequencing, strict sequencing, negative, vertical, ignore, consider, assertion, and loop fragments.

USE CASE DIAGRAM OF LIBRARY MANAGEMENT SYSTEM.



- Expt. No. _____
- Aim: To perform use case analysis
- Theory
 - Actors: Any human involved in the system.
 - System: The environment.
 - Relationship: Shows how the system interacts with each other.
 - use cases: The functions of the system.
- 1) Log in
- Actors: User, Librarian.
 - Description: The process through which a user logs into the system, typically using a user ID and password for security purposes.
- 2) Issue books
- Actors: User, Librarian.
 - Description: The process of lending a book to a user.
- 3) Return books
- Actors: User, Librarian.
 - Description: The process of returning a borrowed book to the library.

Practical-6

Aim: To perform the user's view analysis for the System use case diagram.

Theory:

- Actors: Any human or external system that interacts with the system.
- System: The main component of the system being modeled.
- Relationships: Show how the actors and systems interact with each other.
- use cases: The services the system knows how to perform

1.) Login

- Actors: User, librarian
- Description: The "Login" use case involves the process by which a user gains access to the library management system, typically by providing valid credentials such as a user ID and password. This use case is essential for security purposes and ensures that only authorized individuals can access the system's functionalities and resources.

2.) Issue books

- Actors: User, librarian
- Description: Allow library user to borrow books and librarians to facilitate the book issuance process.

3.) Return books

- Actors: User, librarian

4.) Search a Book

- Actors: user
- Description: Allow library users to search for books based on various criteria.

5.) Add member

- Actors: Librarian
- Description: Allows librarians to add new members(users) to the library Management System.

6.) Delete Member

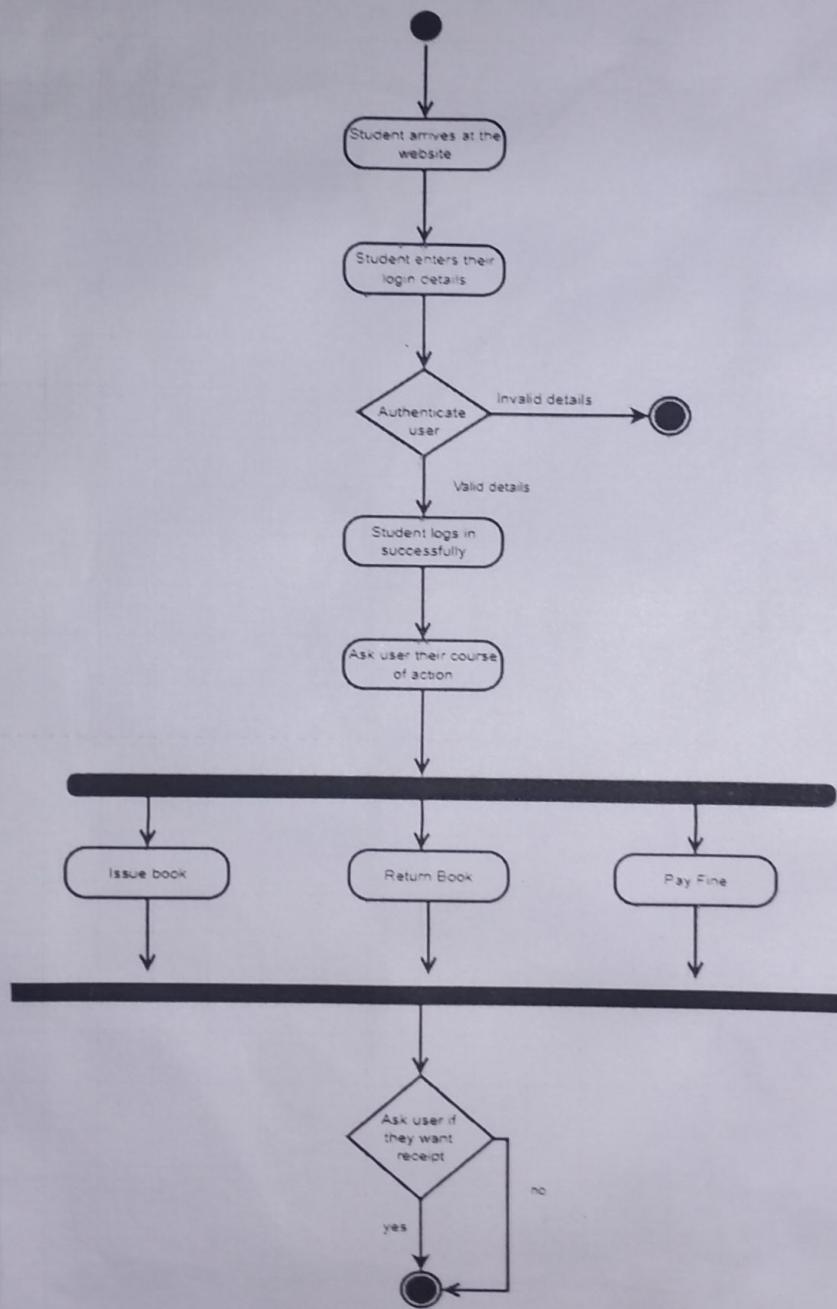
- Actor: Librarian
- Description: Allow librarians to remove existing members (users) from the library management System.

7.) Maintain book records

- Actors: Librarian
- Description: Allows librarians to maintain accurate and up-to-date records of books in the library Management System.

ACTIVITY DIAGRAMS OF LIBRARY MANAGEMENT SYSTEM

LOGIN



Practical-7

- Aim: To draw the behavioral view Diagram: State chart diagram or Activity diagram.

- Theory:-

Activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities. It is a type of behavioral diagram and we can depict both sequential processing and concurrent processing of activities using an activity diagram i.e. an activity diagram focuses on the condition of flow and the sequence in which it happens.

- Components of Activity Diagram:-

- 1) Initial State

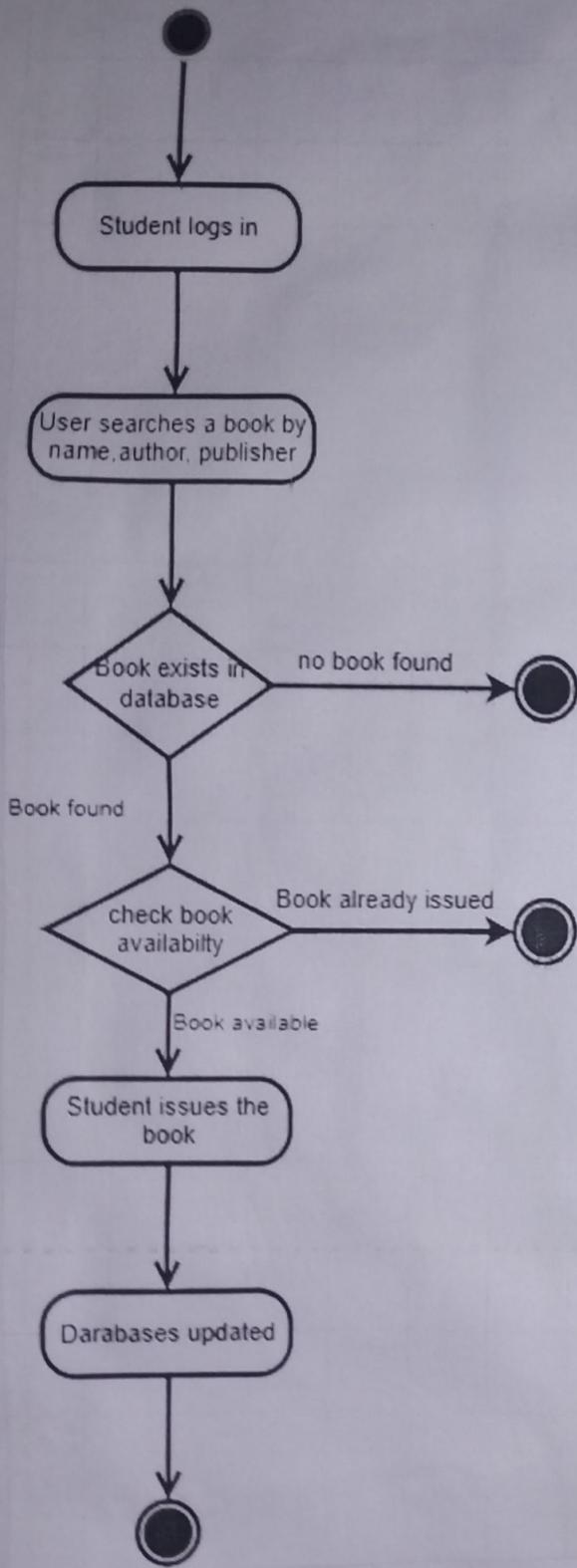
- The starting state before an activity takes place is depicted using the initial state.

- A process can have one initial state unless we are depicting nested activities. We use a black filled circle to depict the initial state of a system.

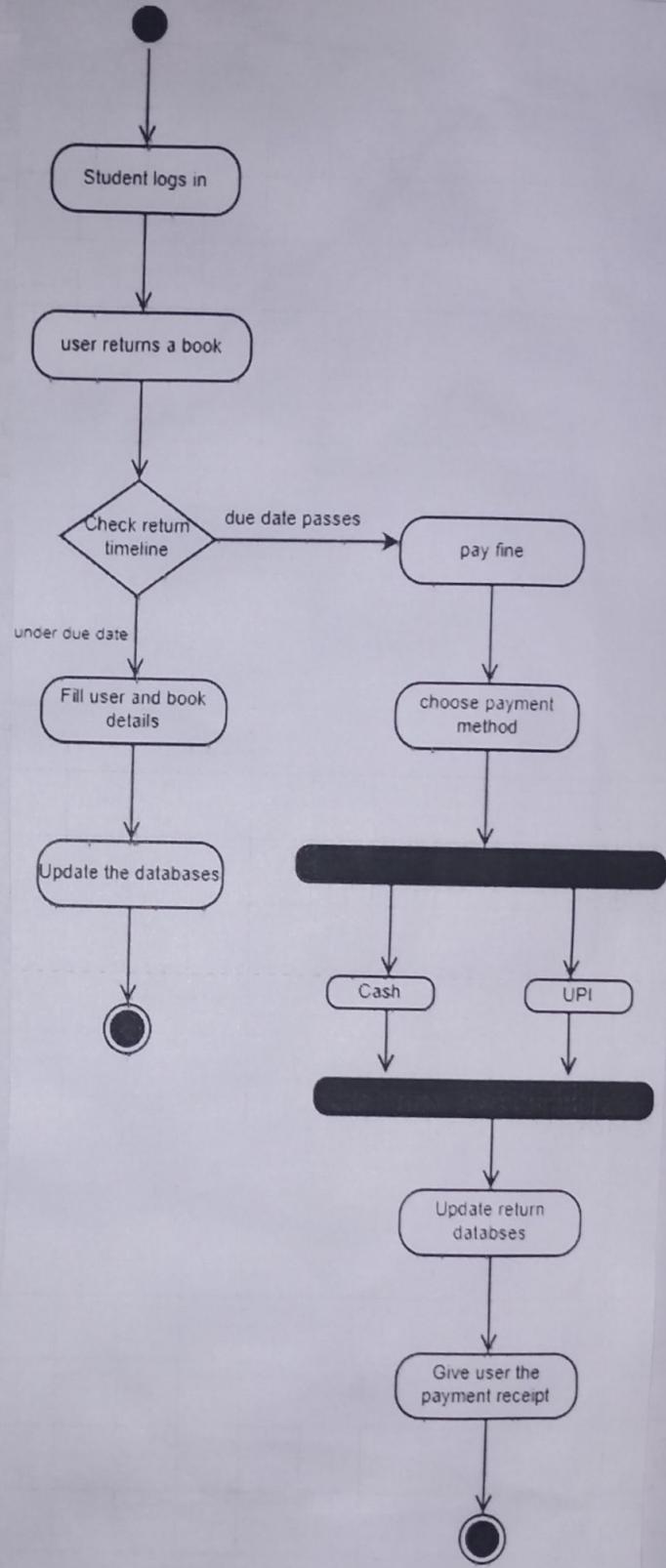
- 2) Action or Activity state:

An activity represents execution of an action on objects or by objects. We represent an activity using a rounded rectangle with rounded corners. Basically any action or event that takes place is represented using an activity.

ISSUE Books



RETURN Books



3.) Action flow or control flows

- Action flows or control flows are also referred to as paths and edges. They are used to show the transition from one activity state to another activity state.
- An activity state can have multiple incoming and outgoing action flows. We use a line with an arrow head to depict a control flow. If there is a constraint to be adhered to while making the transition it is mentioned on the arrow.

4.) Decision node and Branching

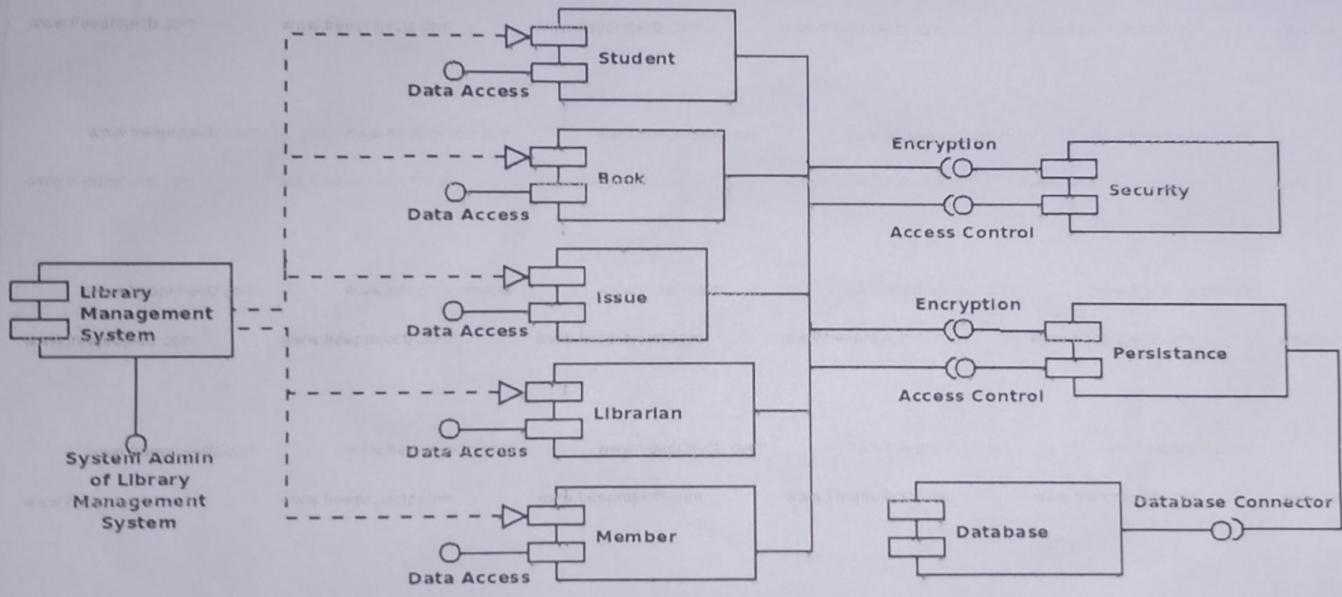
When we need to make a decision before deciding the flow of control, we use the decision node. The outgoing arrows from the decision node can be labelled with conditions or guard expressions. It always includes two or more output arrows.

5.) Fork

- Fork nodes are used to support concurrent activities. When we use a fork node both the activities get executed concurrently.
- We use a rounded solid rectangular bar to represent a fork notation with an incoming arrow from the parent activity state and outgoing arrows towards the newly created activities.

6.) Join

Join nodes are used to support concurrent activities converging into one. For join notation we have two or more incoming edges and one outgoing edge.



Component Diagram of Library Management System

Practical - 8

- Aim:- Draw the component diagram.

Theory

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system.

Components of component diagram

- Issues component
- Student Component
- Librarian Component
- Member component
- Address component

Features of Library Management System Component Diagram

- you can show the models the components of library Management System.
- Model the database schema of library management System.
- Model the executables of an application of library Management System
- Model the system's source code of library management System.

Practical-9

- Aim: Draw the Deployment diagram.

- Theory:

A deployment diagram illustrates how software architecture, designed where the software will run as nodes. It maps out the deployment of software components onto hardware nodes and depicts their relationships through communication paths, enabling a visual representation of the software's execution environment across multiple nodes.

- Key Elements of a Deployment Diagram

- 1) Nodes

These represent the physical hardware entities where software components are deployed, such as servers, workstations, routers, etc.

- 2) Components: represents software modules or artifacts that are deployed onto nodes, including executable files, libraries, databases etc.

- 3) Artifacts: physical files deployed onto nodes, embodying the actual implementation of software components, such as executables, scripts, databases etc.

- 4) Dependencies: reflects relationships or connections between nodes and components, indicating communication paths, deployment constraints, or other dependencies.

5.) Associations

Show relationships between nodes and components, signifying that a component is deployed on a particular node, thus mapping software components to physical nodes

6.) Deployment Specification

Describes the configuration and properties of nodes and components, encompassing hardware specifications, software configurations, communication protocols etc.

7.) Communication paths

Represent channels or connections facilitating communication between nodes and components, including network connections, communication protocols, etc.