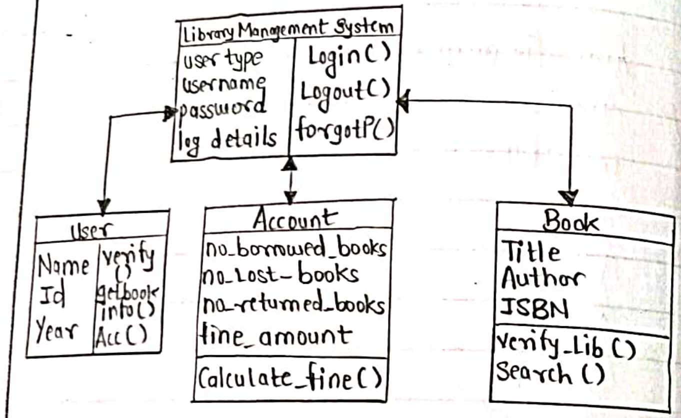
Experiment No. 1

Date: 8/1/25

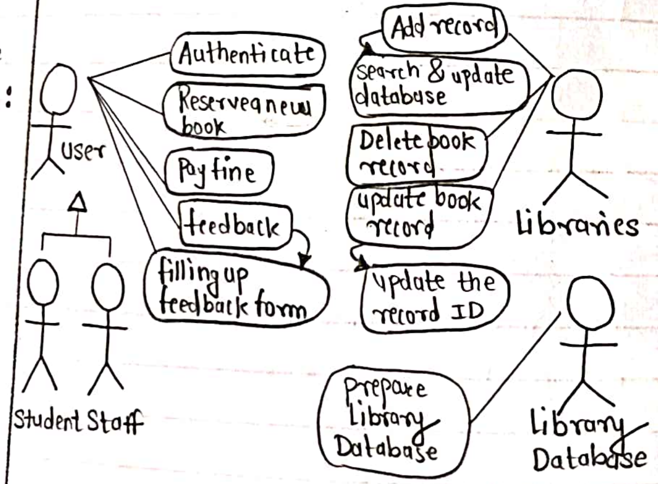
Aim: To implement different UML diagrams along with their diagramatic representation.

Objective: To understand & apply various UML diagrams for visualizing & modelling a software system.

Class Diagram:



Use Case Diagram:

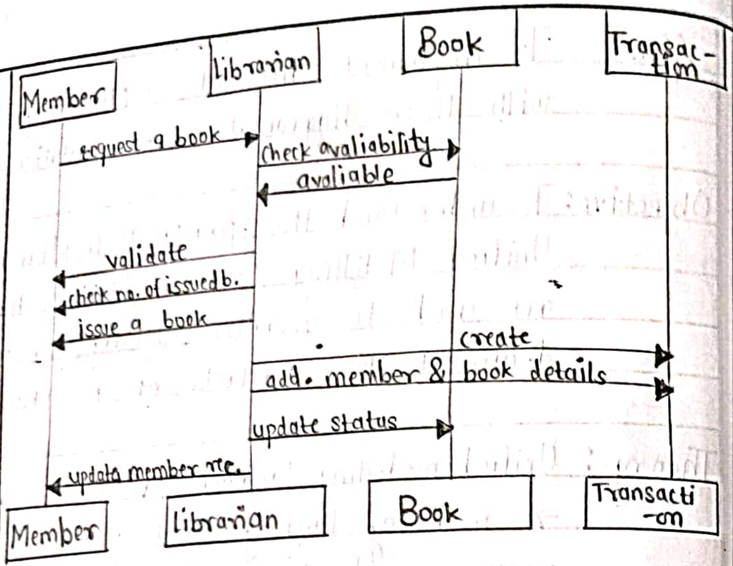


Experiment No. 1

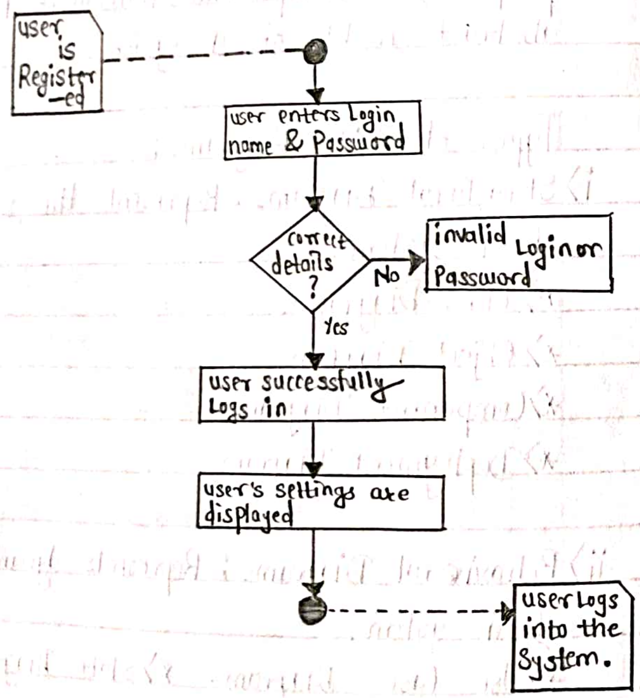
2

Aim:	To implement different UML diagrams along with their diagramatic representation.
Objective:	To understand the practical implementation of Unified Modelling Langauge (UML) diagrams, which are used to visualize, specify, construct and document the artifacts of a software system.
Theory:	Unified modelling langauge [UML] is a standardize modelling Langauge that provides a general-purpose developmental framework for creating abstract models of a system.
	Types of UML Diagrams:
i)	Structural Diagrams: Represent the static aspects of a system
	*> Class Diagram
	*> Object Diagram
	*> Component Diagram
	*> Deployment Diagram
ii)	Behavioral Diagrams: Represents dynamic aspects of a system
	*> Use Case Diagram. *> State Diagram
	*> Sequence Diagram
	*> Activity Diagram.

Sequence Diagram:



Activity Diagram:



user is Registered
user enters Login name & Password
correct details ?
No
invalid Login or Password
Yes
user successfully Logs in
user's settings are displayed
user Logs into the System.

Each Diagram serves a different purpose in modelling a software system

Class Diagram: Identify system entities (classes) & their relationships. Adds attributes, methods & connections like association or inheritance.

Use case Diagrams: Define actors & their interactions with the system. link actors to the use cases

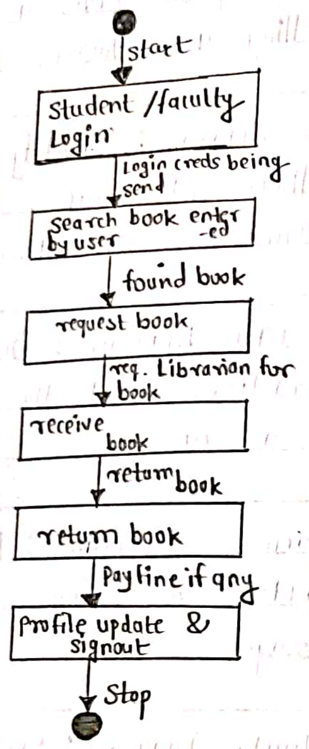
Sequence Diagrams: Maps interactions between objects over time. Represents lifelines & messages.

Activity Diagram: Show workflows & activities. Use transitions & decision nodes.

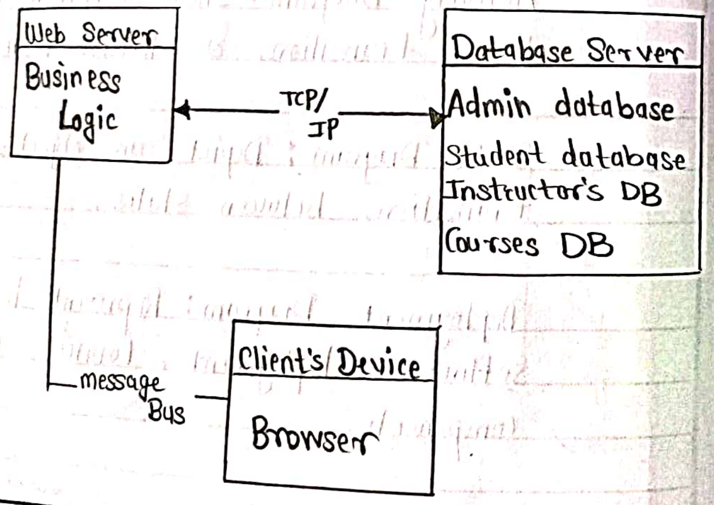
State Diagram: Depict an object's state and transitions between states.

Deployment Diagram: Represent hardware & software deployment. Connect nodes & it's components.

State Diagram:



Deployment Diagram:



Conclusion: The experiment demonstrated the practical application of UML diagrams to analyze & model software systems effectively.

Example: System: Library management System (LMS):
Use class diagram for entities like "Book" & "User".
Use class diagram for interactions like "borrowing books"
Sequence Diagram for "issue workflow".

Result: Different UML diagrams were implemented successfully, acheiving the desired system's representation.

Conclusion: The experiment demonstrated the practical application of UML diagrams to analyze & model software systems effectively.

[illegible]

④