

## Experiment-1

### Rational Rose:

Rational Rose is a legacy software design tool from IBM used for visual modelling and component construction of enterprise-level software application. It supports various languages like UML and helps in designing and maintaining object-oriented software.

### Key features.

- Supports multiple programming languages
- Provides round-trip engineering
- Integrates with other IBM Rational Rose tools.
- Allows for the creation of UML diagrams like usecase diagrams, class diagrams, sequence diagrams etc.

### StarUML

StarUML is an open source software modelling tool that supports the creation of UML diagrams. It is designed to be lightweight and fast providing a robust environment for creating detailed and extensive UML models.

### Key features.

- Supports multiple UML 2.x diagrams
- Extensible with plugins
- Realtime collaboration through cloud services
- code generation and reverse engineering capabilities
- user friendly interface with drag and drop features.

### Umbrello:

Umbrello is a UML modeling tool that is part of the KDE software compilation. It allows user to create diagrams for software design and other modelling needs.

#### Key features

- supports all standard UML diagrams
- provides code generation for various programming languages
- open source and cross platform
- simple and intuitive interface
- support xMI for model interchange.

### Visual paradigm

Visual Paradigm is a sophisticated modelling tool that supports UML and others standards like BPMN. It is widely used for both software development and business modelling.

#### Key features

- comprehensive support for UML, BPMN, ERD, and other diagrams
- Integration with various IDEs
- Agile and scrum supports
- collaboration tool for team based modelling
- visual modelling with drag and drop and automated layout features.

### Microsoft Visio:

Microsoft Visio is a versatile diagrammatical tool from Microsoft that supports a wide range of diagram types, including flowcharts, network diagrams, organizational charts and

UML diagrams.

### Key features

- Extensive technique template library for various types of diagrams
- Integration with Microsoft Office suit
- Realtime collaboration and sharing features
- Easy to use with drag and drop functionality
- Support both simple and complex diagramming needs

Comparison and usecases.

#### 1. Rational Rose

Best for large enterprise with complex systems needing integration with other IBM tools.

#### 2. Star UML

Suitable for developers looking for an open source, lightweight and fast UML modelling tool.

#### 3. Umbrello

Idea for KDE users and those looking for an open source alternative with basic UML modelling needs

#### 4. Visual Paradigm

Great for comprehensive modelling needs, including software development and business processes with extensive collaboration features.

#### 5. Microsoft Visio

Perfect for users who need a versatile diagramming tool that integrates well with Microsoft Office products and can be used for a variety of diagram types beyond just UML.

## Experiment-2

Aim: Understanding different views that the UML aims to visualize through different modelling diagrams.

User's view: Usecase diagram, Structural view: class Diagram, Object Diagram and Behavioural view: Sequence diagram, Collaboration diagram and Statechart diagram, Activity diagram, Environmental view: Deployment diagram  
Implementation view: Component Diagram

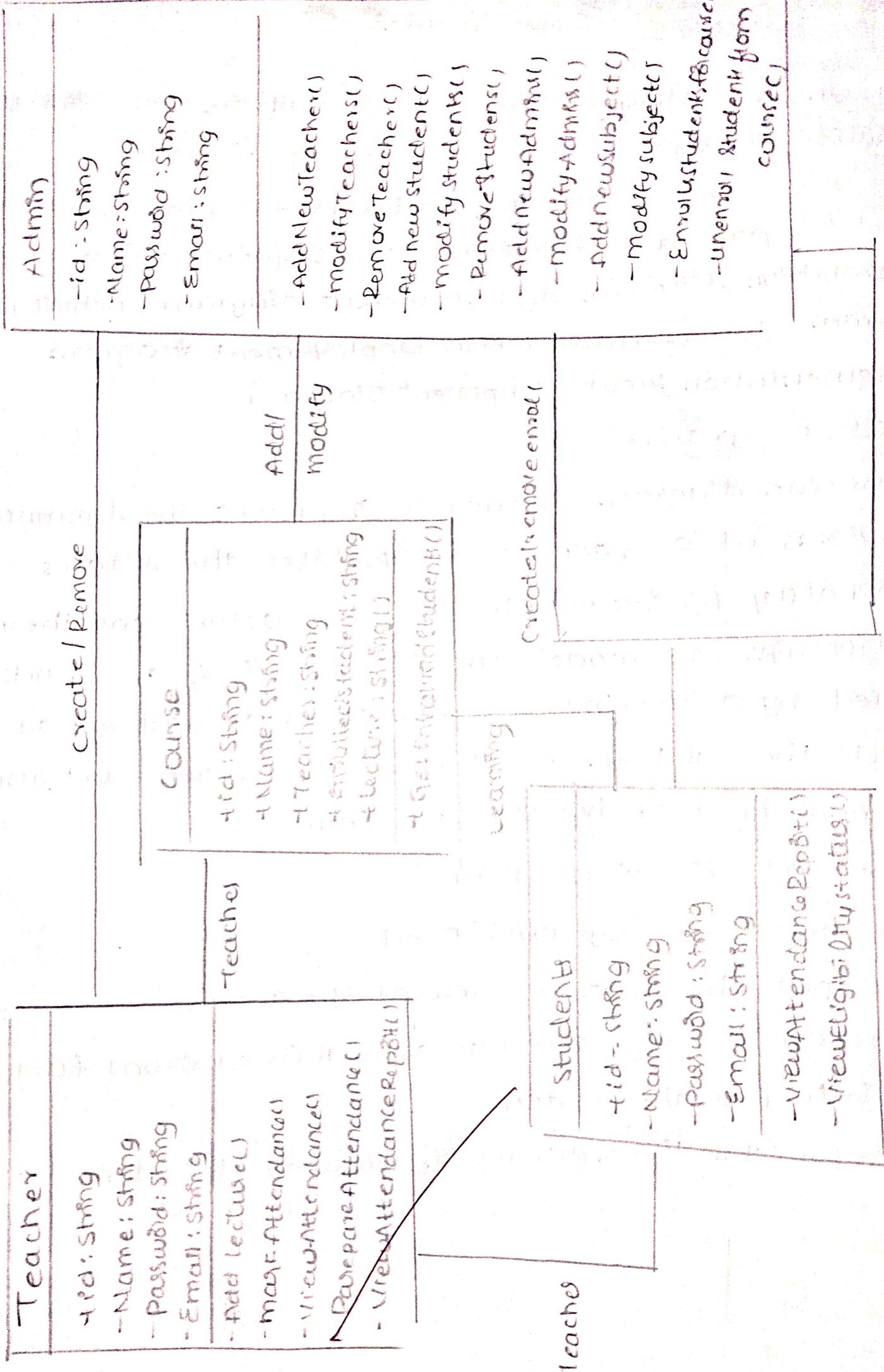
### usecase diagram:

A Usecase diagram is used to represent the dynamic behaviours of a system. It encapsulates the systems functionality by incorporating usecases, actors and their relationships. It models the tasks, services, and functions required by a system / subsystem of an application. It depicts the high level functionality of a system and also tell how the user handles a system

### Purposes of usecase diagram

1. It gathers the system's need
2. It depicts the external view of the system
3. It recognizes the internal as well as external factors that influence the system
4. It represents the interaction between the actors.

# class diagram + to Attendance Management System



## Class Diagram

The class Diagram depicts objects a static view of an application. It represents the types of objects residing in the System and the relationships between them. A class consists of its Objects and also it may inherit from other classes. A class Diagram is used to visualize, describe, documents various different aspects of the system and also construct executable software code.

### Purpose of class Diagram

- 1- It analyses and designs a static view of an application
- 2- It describes the major responsibilities of a system.
- 3- It is a base for component and deployment diagrams
- 4- It incorporates forward and reverse engineering.

### Key elements of class Diagram

- 1- Class
- 2- Attributes
- 3- Methods
- 4- Relationships

### Advantages of class Diagram

- clear Representation of system structure.
- Encourage modularity
- Facilities of object oriented design
- Effective communication tool
- Simplifies complex system
- Supports reusability
- Improves system maintainability

• mailid : sukh@smail.com

• Name : suha

• StaffId : H10001

• Staff2 : staff2

• Year : 1111

• Semester : V

• Course : Physics

• Name : Madhu

• Student2 : student

• Subject : English

• Subject2 : 92

• Subject3 : 69

• Subject4 : 62

• Subjects : 90

• Subjects2 : 92

• Subjects3 : 92

• Subjects4 : 92

• Courses : CS

• Semester : VI

• Year : 11

• Student : student

• Staff : staff

• Staff2 : staff

• Staff3 : staff

• Staff4 : staff

• Dept : CSE001

• Venue : ABLOCK

• Department : Department1

• StaffName : Walia

• Dept : CSE

• Semester : VII

• Year : 1183

• College : Organisation

• CollegeName : VUIT

• Address : P.D

• Phone : 011126109

• City : DENT : 20

• CityOfCollege : 30

• Avenue : BELBLOCK

• Department : Department3

• Dept : CHEMISTRY

• DeptId : CHEM01

• Venue : CSEBLOCK

• Department : Department4

• DeptId : H10001

• Venue : 2 BLOCK

• Semester : VII

• Year : 1111

## Object Diagram.

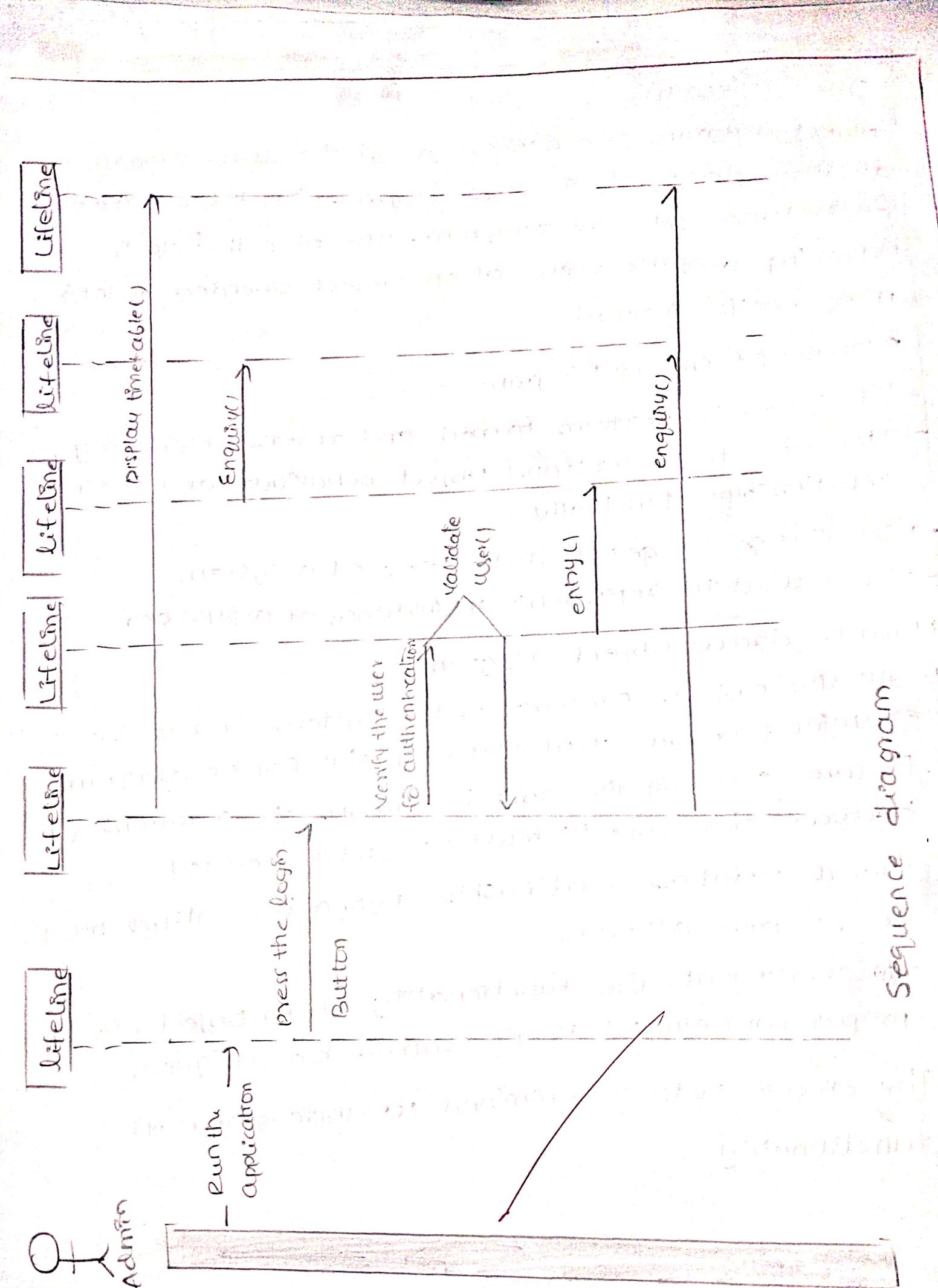
Object diagram are dependent on the class diagram as they derived from class diagram. It represents an instance of class diagram. The objects help in portraying a static view of an object oriented system at a specific instant.

### PURPOSE OF OBJECT DIAGRAM.

- It is used to perform forward and reverse engineering.
- It is used to understand Object behaviour and their relationships practically.
- It is used to get a static view of a system.
- It is used to represent an instance of a system.

### HOW TO DRAW OBJECT DIAGRAM.

1. All the objects present in the system should be examined before start drawing the object diagram.
2. Before creating the object diagram, the relations between the objects must be acknowledged.
3. The association relationship among the entities must be cleared already.
4. To represent the functionality of an object, a proper meaningful name should be assigned.
5. The objects are to be examined to understand its functionality.

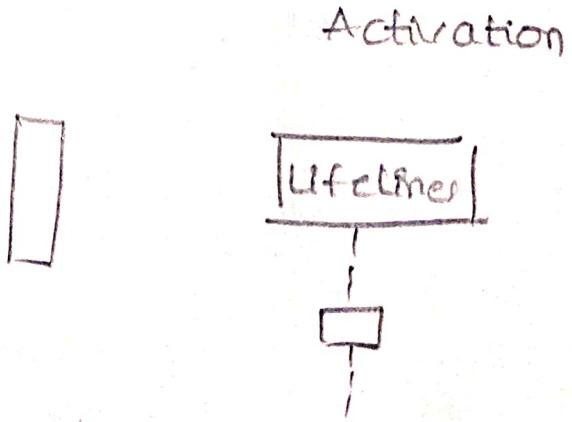
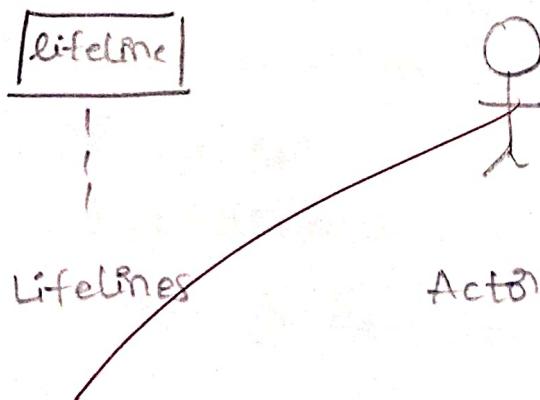


## Sequence Diagrams

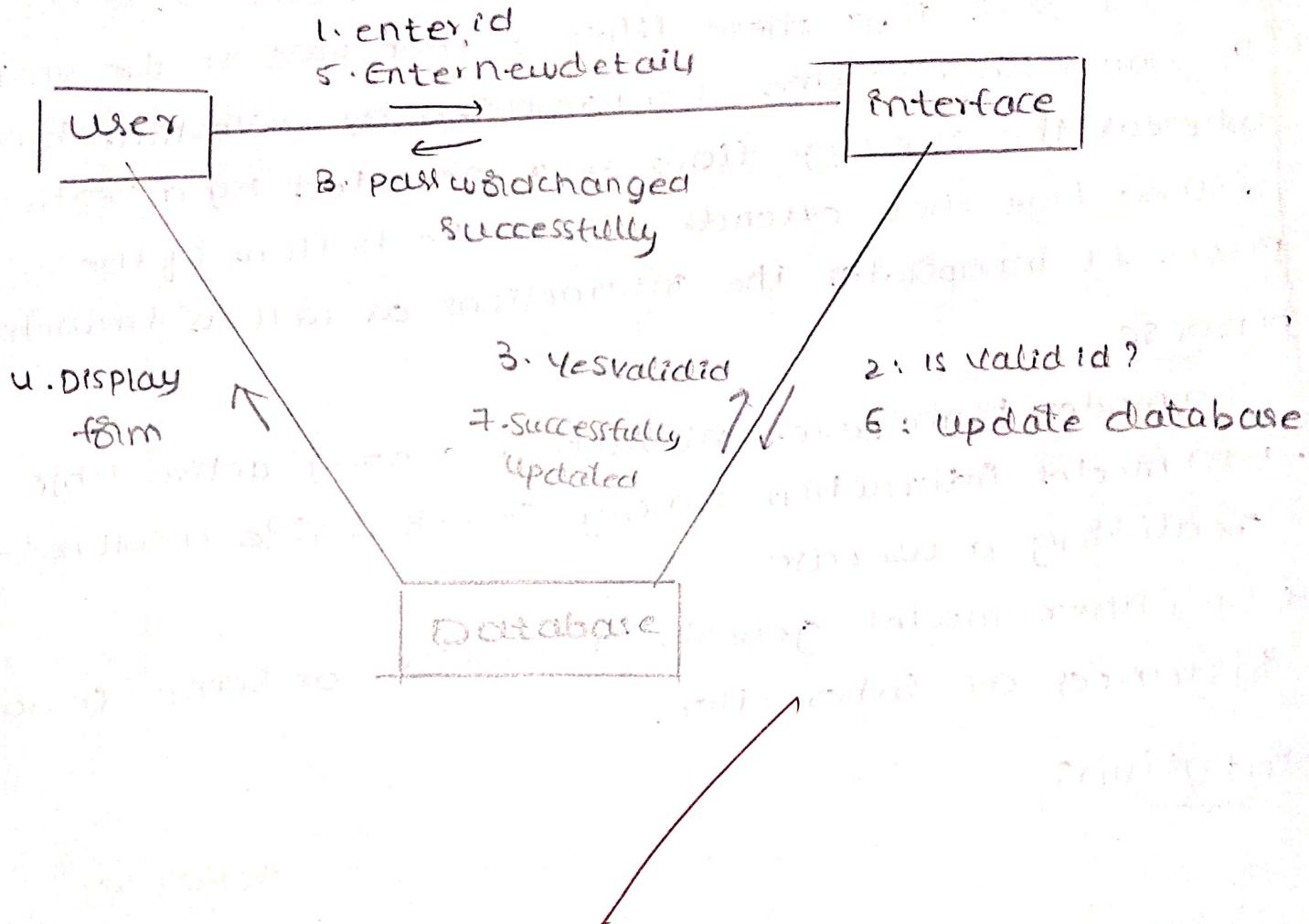
The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines at a time. Ordered sequence of events, such that these lifelines took part at the same time. In UML, the life line 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 interactions as well as branching purpose.

- To model high level interaction among active objects
- To model interaction among objects inside a collaboration realizing a use case
- It either models generic interaction or some certain instances of interaction.

## Notations



## collaboration Diagram



## Collaboration Diagram

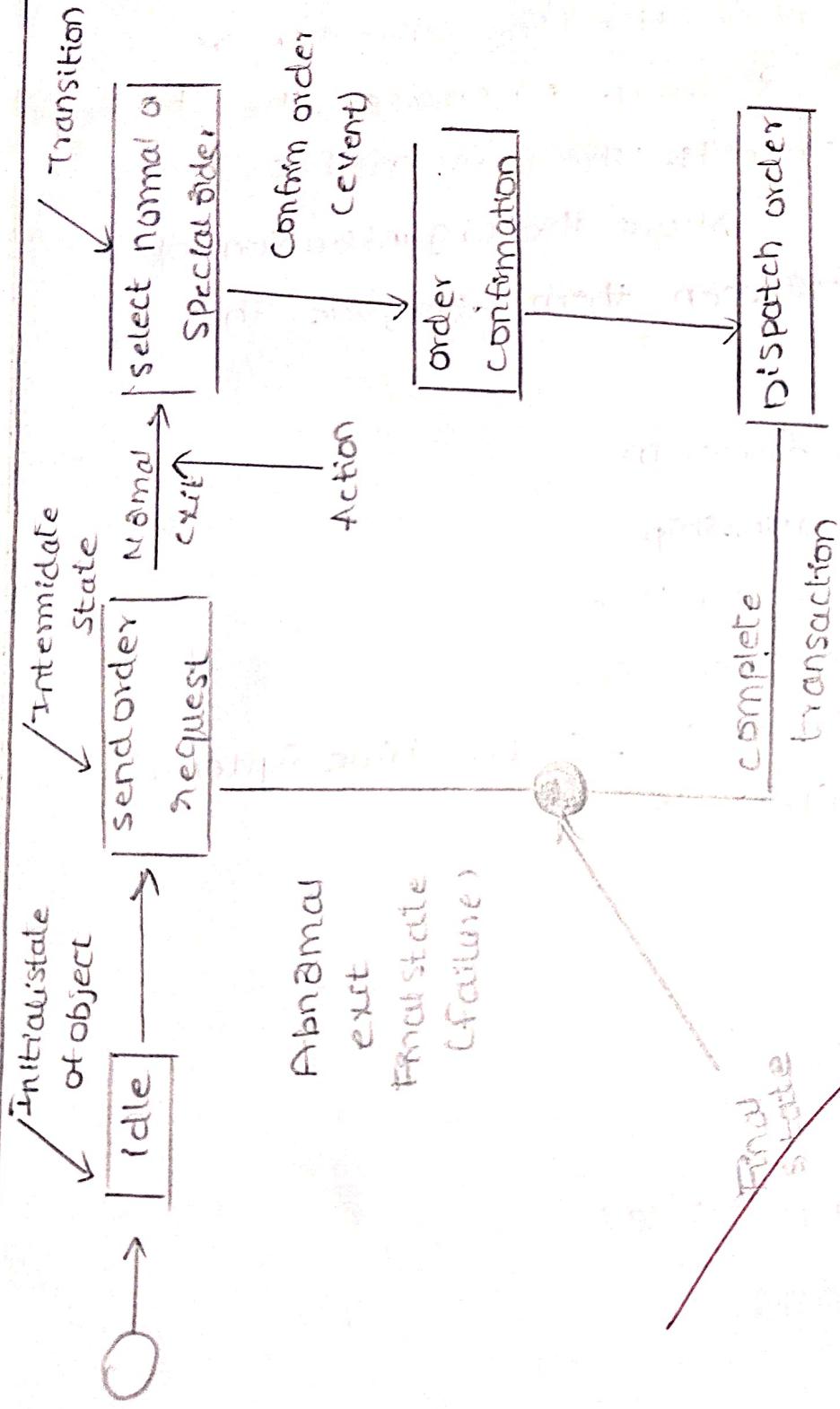
A collaboration diagram also known as a communication diagram in UML is a type of interaction diagram that focuses on how objects or components interact to fulfill a particular task or process. Unlike sequence diagram, which emphasize the time ordering of messages, collaboration diagram emphasize the structural relationships between objects. The main goal of a collaboration diagram is to show the organization of objects and the links between them, alongside the messages exchanged.

Purpose of collaboration diagram.

- Emphasizing Object Relationships.
- Visualizing Object collaboration
- Representing Real-world scenarios
- Representing object interaction in Real time Systems
- Designing system Architecture

Advantages

- Focus on structure
- Compact Representation
- Object-Centric view
- Clear visualisation of object roles
- Useful for Reverse engineering



State chart diagram for an order management system

## Statechart Diagram

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes the different states of a component in a system. The states are specific to a component object of a system.

A statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

### Purpose of statechart diagram.

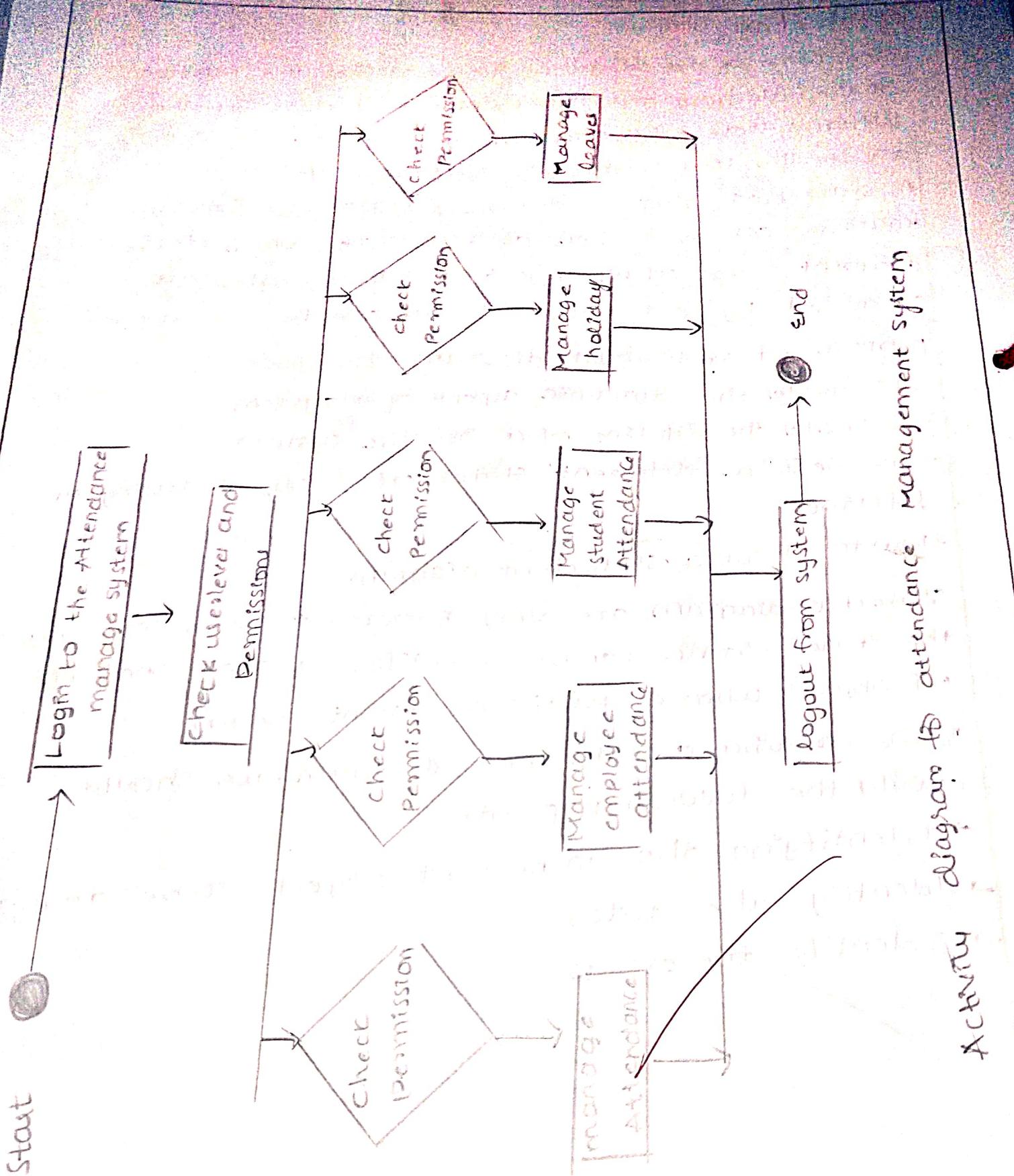
- TO model the dynamic aspect of a system
- TO model the lifetime of a reactive system
- TO describe different states of an object during its lifetime.

### How to draw a statechart diagram.

Statechart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs.

Before drawing a statechart diagram, we should clarify the following points.

- Identifying the important objects to be analyzed
- Identify the states
- Identify the events.



management system

attendance

activity

activity

## Activity Diagram

Activity Diagram are used to illustrate the flow of control in a system and refer to the steps involved in the execution of usecase - we can depict both sequential processing and concurrent processing of activity using an activity diagram re an activity diagram focuses on the condition of flow and the sequence in which it happen

Purpose of an Activity Diagram

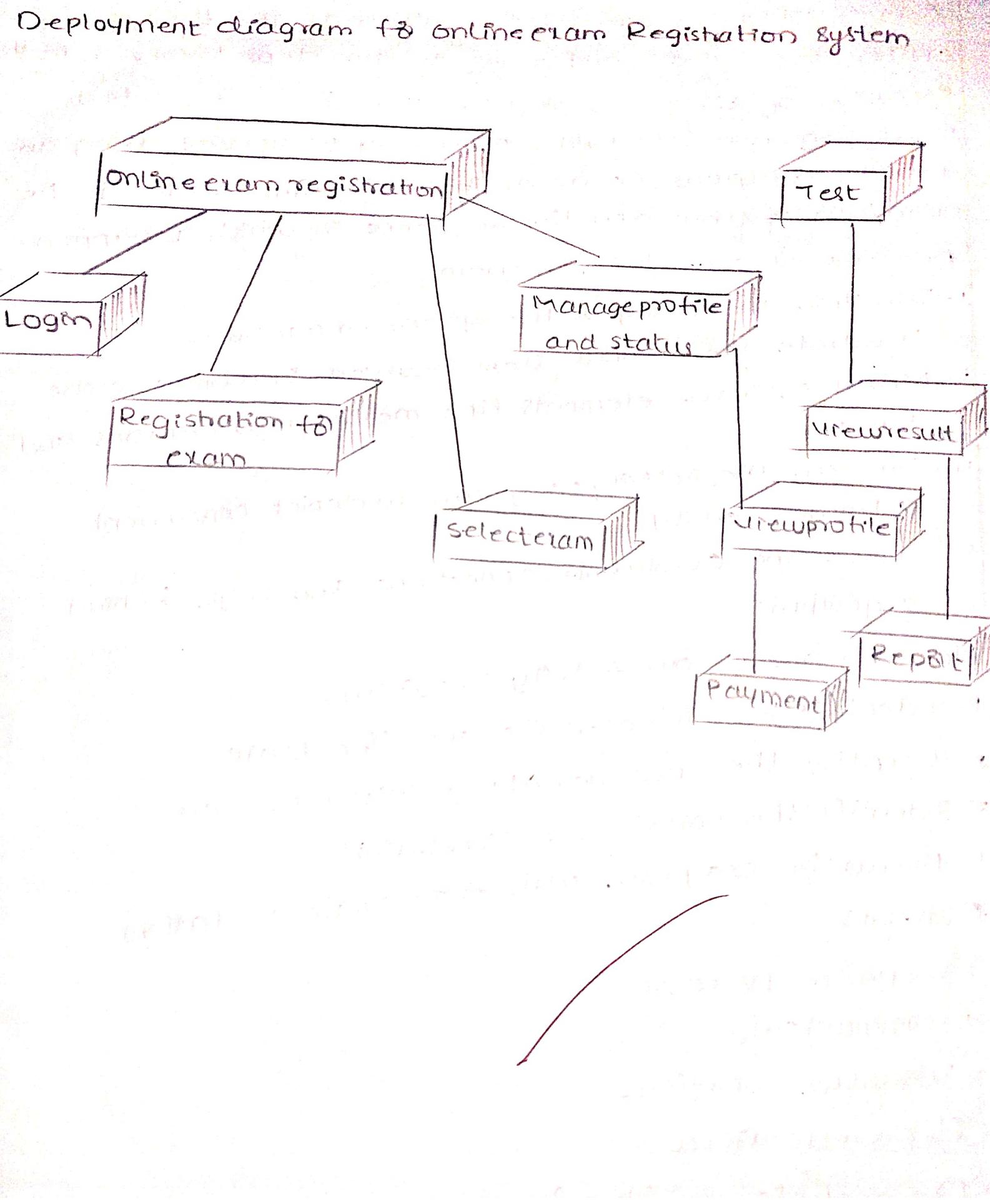
1. Dynamic modelling of the system or a process
2. Illustrate the various steps involved in a UML usecase
3. Model software elements like methods, operations and functions
4. We can use Activity Diagrams to depict concurrent activities easily
5. Shows the constraints, conditions and logic behind algorithms.

How to draw an Activity Diagram.

1. Identify the initial state and final states.
2. Identify the intermediate activities needed.
3. Identify the conditions & constraints
4. Draw the diagram with appropriate Notations

## Benefits

- Improve processes
- communicate
- visualize workflows
- Represent usecase
- Demonstrate algorithm logic



## Deployment Diagram

A deployment diagram illustrates how software architecture designed on a conceptual level is translated into the physical system architecture where the software will run as nodes. It maps out the deployment of software components onto hardware nodes & depicts their relationships through communication paths, enabling a visual representation of the software's execution environment across multiple nodes.

### Purpose of deployment diagram

- show how software and hardware interact
- Identify which software is deployed by which hardware
- Show the processing runtime for hardware
- Assess the implications of resource allocation.

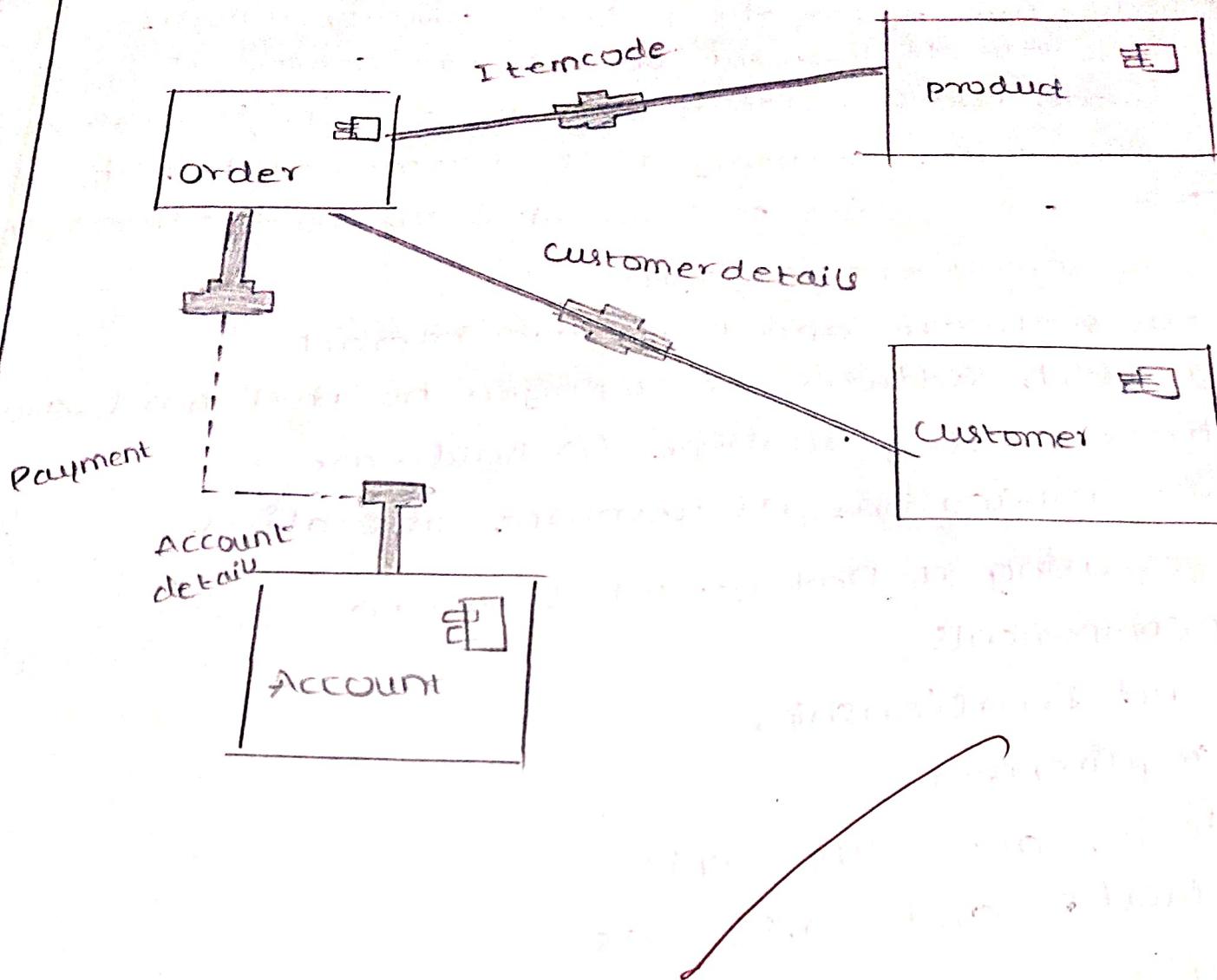
### Steps for creating a Deployment diagram

1. Identify components
2. Understand Relationships
3. Gather requirements
4. Draw nodes and components
5. Connect nodes and components
6. Add details
7. Documentation

### Benefits of Deployment diagram

Used for visualizing, specifying, and documenting embedded client/server and distributed systems and also for managing executable systems.

## Component Diagram for order management system



## Component Diagram

A component-based diagram often called a component diagram, is a type of structural diagram in the unified modelling language (UML) that visualizes the organization and interrelationships of the components within a system.

### Purpose of component Diagram

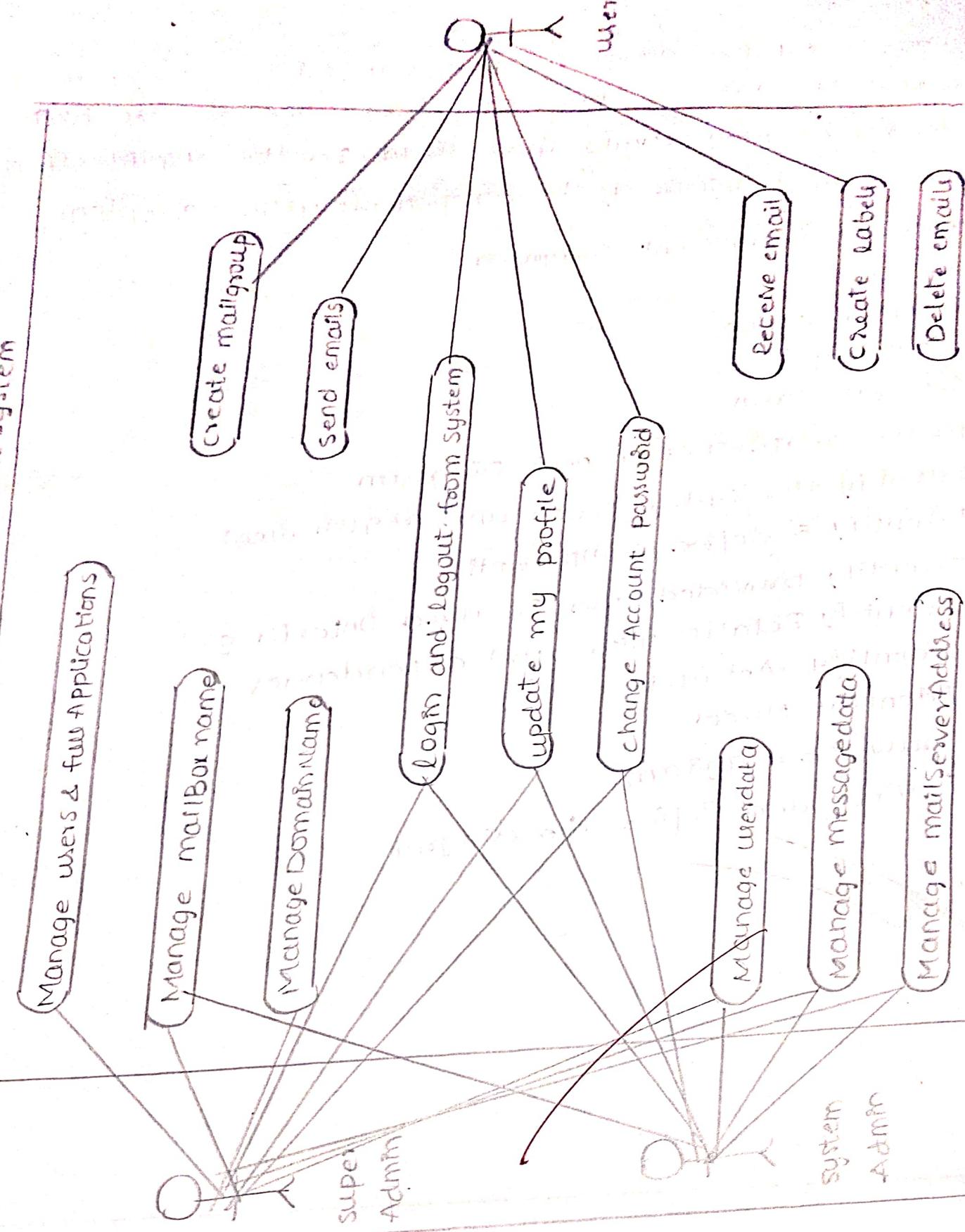
1. Modelling
2. Communication
3. Planning
4. Design Flaws

### Steps to create a component-Diagram.

1. Identify the system scope and requirements.
2. Identify & define components
3. Identify provided and required interfaces
4. Identify Relationships and dependencies
5. Identify Artifacts
6. Identify Nodes
7. Draw the diagram
8. Review and Refine the diagram



## Use case diagram for email client system



### Experiment-3

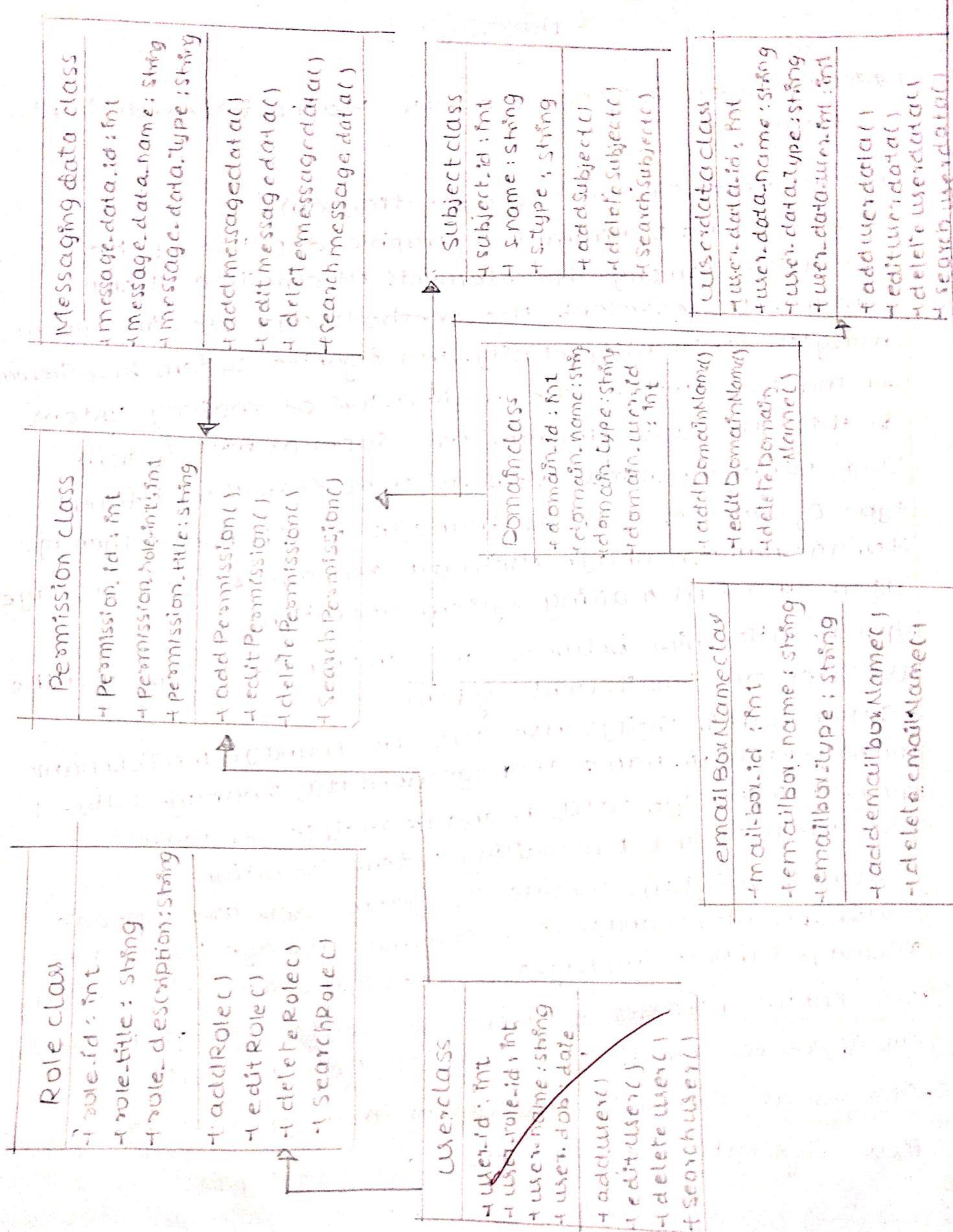
Aim: Create a complete UML Model for email client system.

#### Email client system use case diagram

The use case diagram is a graphic depiction of the interactions among the elements of emailing client system. It represents the methodology used in system analysis to identify, clarify and organize system requirements of mailing system. The main actors of mailing system in this use case diagram are super admin, system user, users, anonymous users, who perform the different type of usecases such as manage mailboxname, manage domain name, manage userdata, manage subject, manage users and full mailing system operations.

#### The Relationships between and among the actors and the usecases of email client system

- Super Admin Entity: usecases are manage mailboxname, manage domain name, manage userdata, manage subject, manage message data, manage mail server address, manage users and full mailing system operations.
- System User Entity: usecases of system users are manage mailbox name, manage domain name, manage userdata, manage subject, manage message data, manage mail server address.
- User Entity: usecases of users are create mail groups, send emails, receive email, create labels, Delete emails.
- Anonymous User Entity: usecases of anonymous users are view information, Fill contactus, Search content



class diagram of email client system.

E-mail client system class diagram describes the structure of a e-mail client system classes, their attributes, operations (or methods) and the relationships among objects. The main classes of the e-mailing system are mailbox name, Domain name, user data, subject, message data, mailserver Address.

classes of email client system class diagram

1. Mailbox name class
2. Domain Name class
3. Userdata class
4. Subject class
5. Message data class
6. Mail Server Address class.

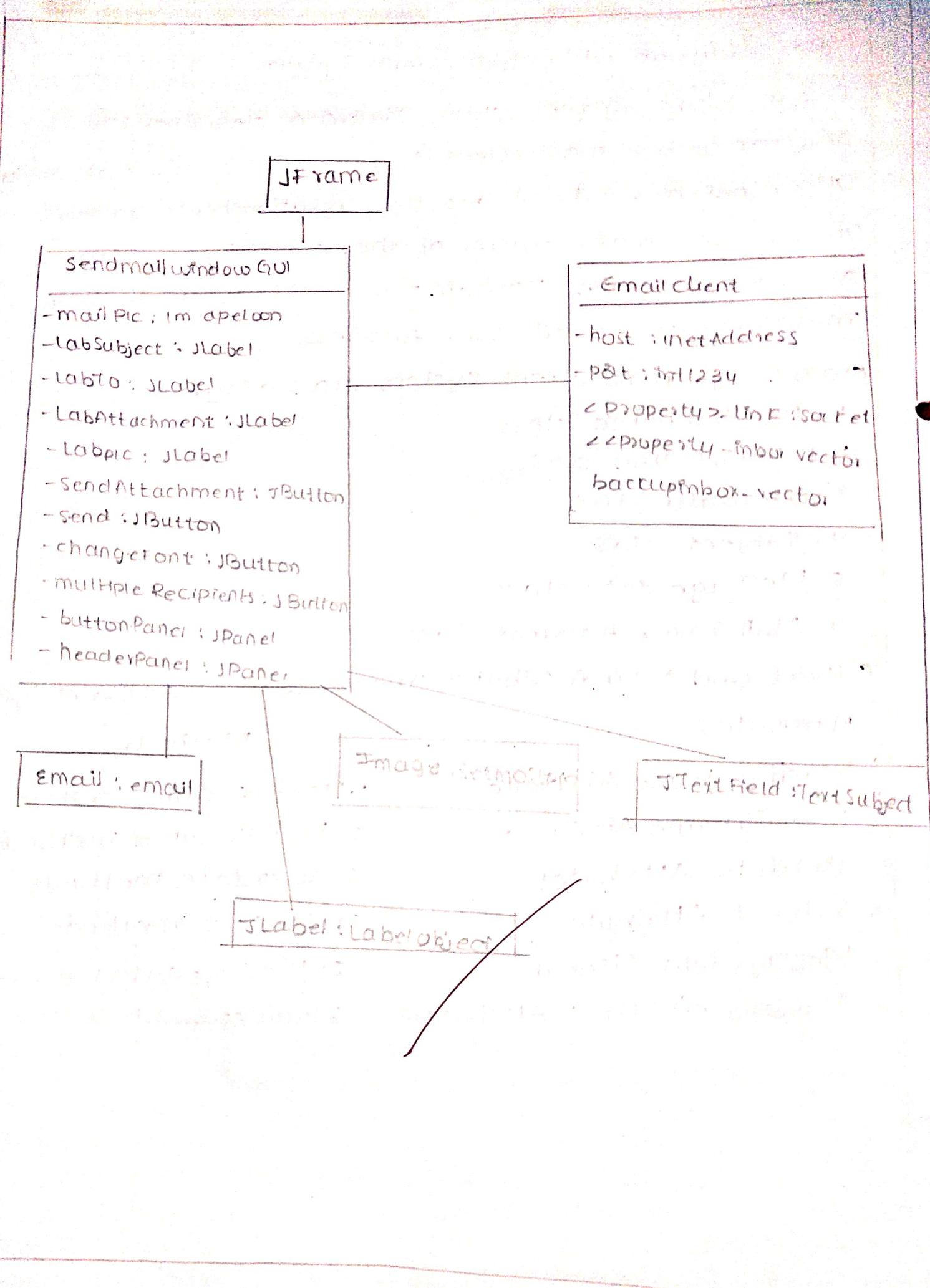
classes and their attributes, methods of email client System

Attributes

1. Mailbox name Attributes
2. Domain name Attributes
3. Userdata Attributes
4. Subject Attributes
5. Message data Attributes
6. Mailserver Address Attributes

Methods

1. Mailboxname methods
2. Domainname methods
3. Userdata methods
4. Subject methods
5. Message data methods
6. MailserverAddress methods



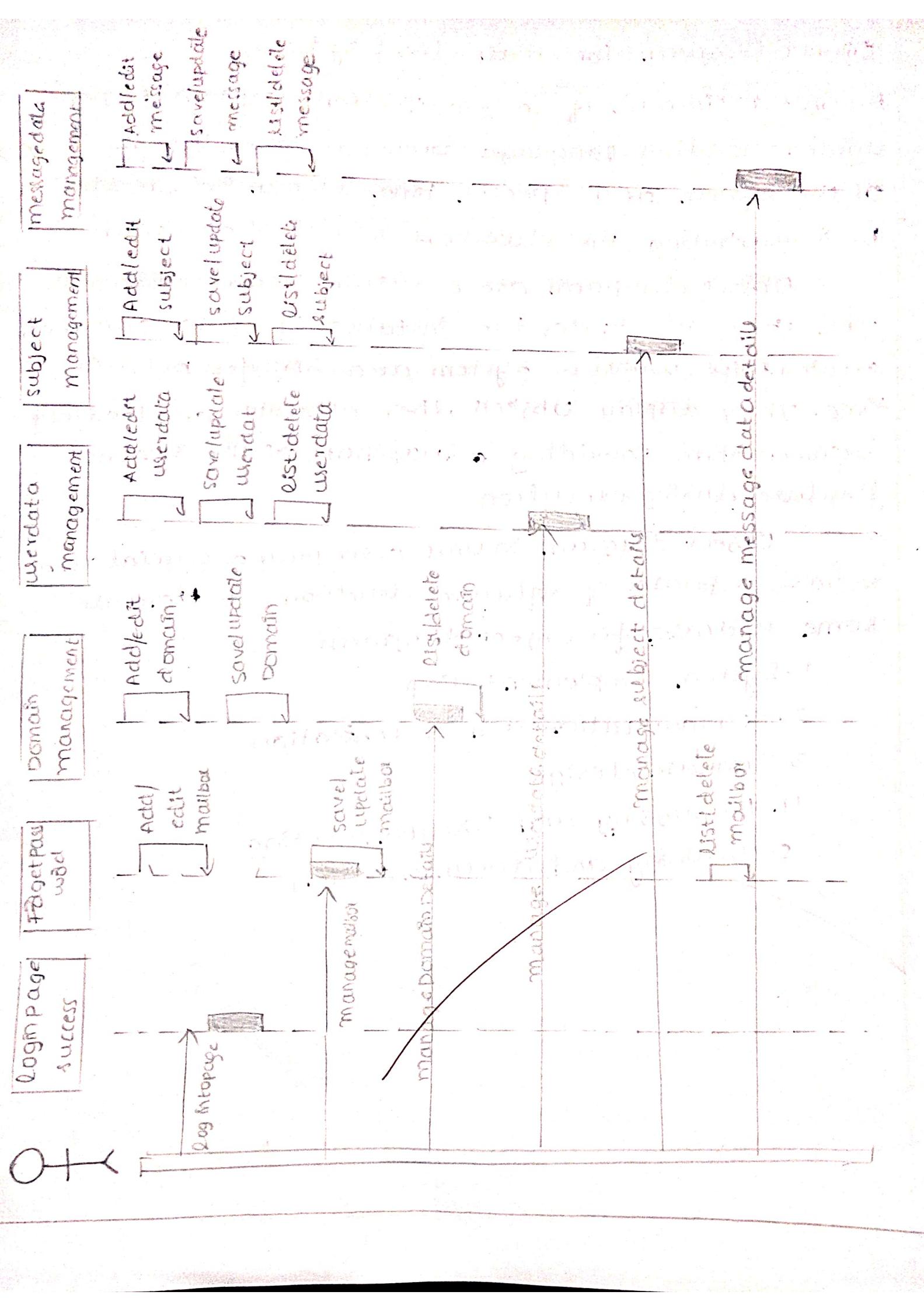
Object Diagram for email client system.

An object diagram of an email client system in the unified modeling language (UML) can show the structure of the system at a specific time. It can be created by instantiating the classifiers in a class diagram.

Object diagrams are a visual representation in UML that illustrates the instances of classes and their relationships within a system at a specific point in time. They display objects, their attributes and the links between them providing a snapshot of the system's structure during execution.

Object diagrams in UML also play a crucial role in various phases of software development. Here are some usecases for object diagrams:

1. System Implementation
2. Communication and collaboration
3. Testcase design
4. Debugging and Troubleshooting
5. Training and documentation



## Sequence Diagram of email client system

This is the UML sequence diagram of e-mail client system which shows the interaction between the objects of mailbox, mailserverAddress, subject, userdata domain. The instance of class objects involved in this UML sequence diagram of e-mail client system are

- Mailbox object
- MailserverAddress object
- Subject object
- \_userdata object
- Domain object

A Sequence diagram for an email client system typically involves interactions between user, emailclient, email server and possibly other components like spam filters or databases. Here's a general structure of how it would look.

Actors / Objects.

1. User

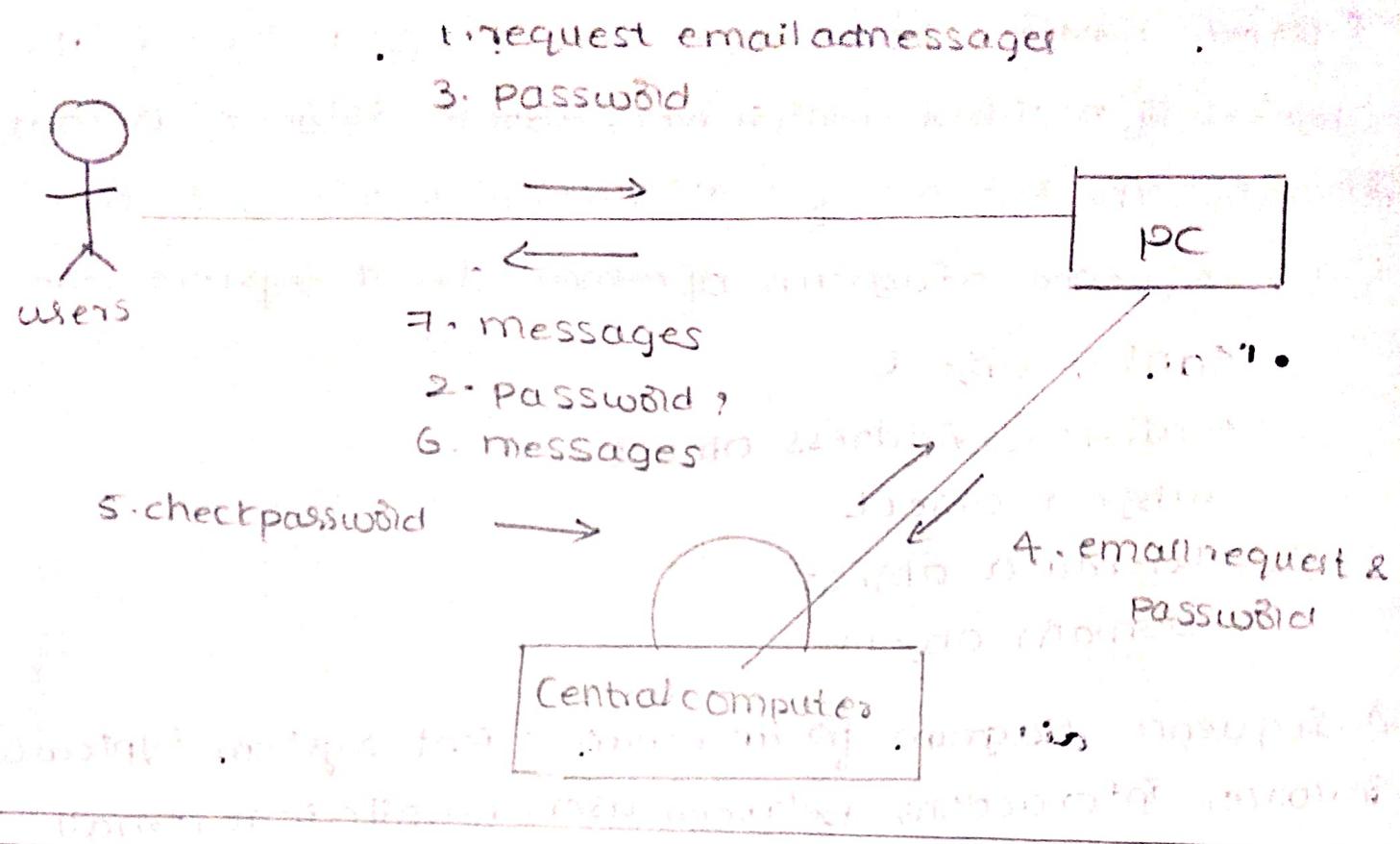
2. Email client (Application)

3. Email Server (SMTP for sending, IMAP/POP3 for receiving)

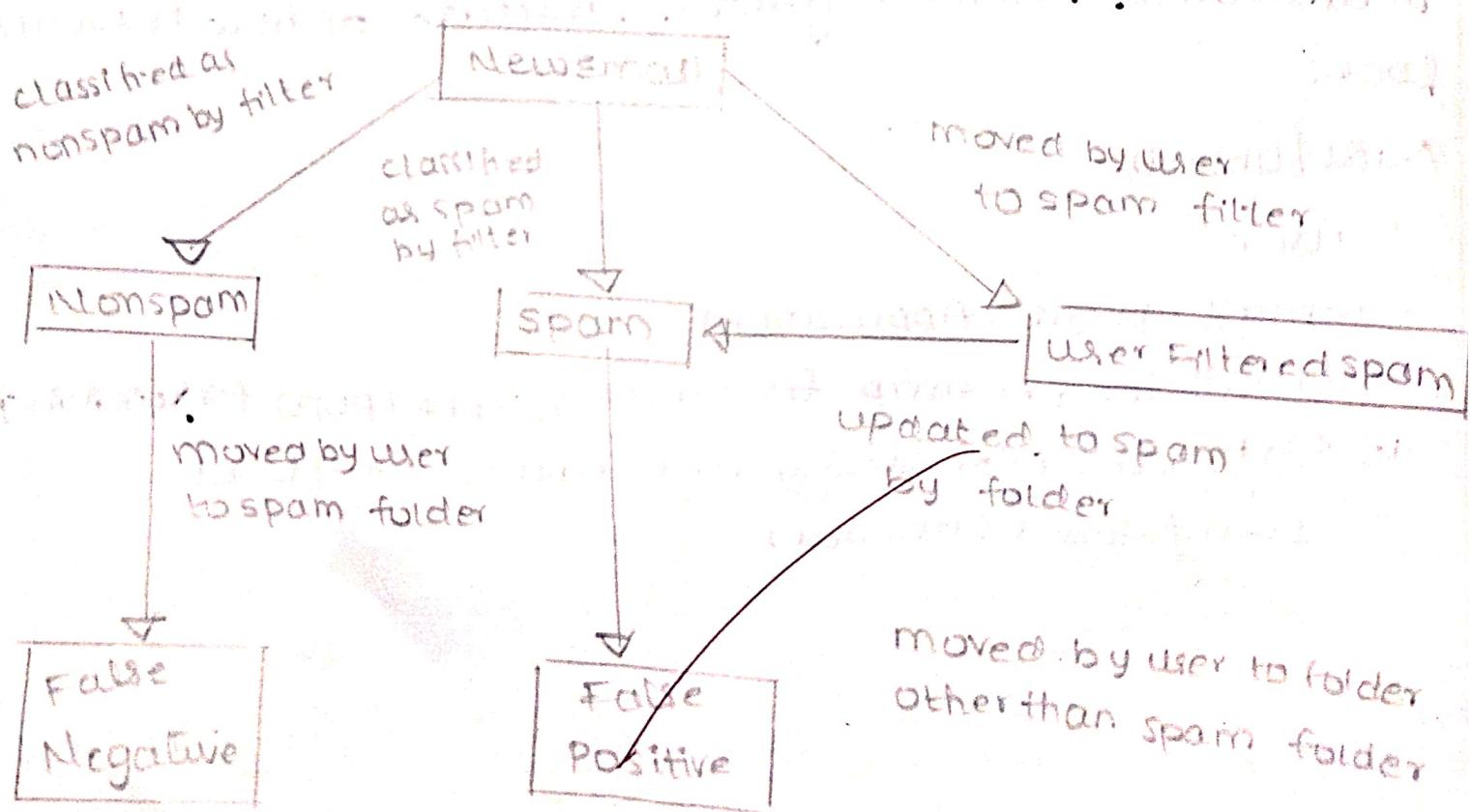
4. Database (for storing sent emails, drafts etc)

5. Spam filters (optional)

# Collaboration diagram of email client system



## Statechart diagram of email client system



## Collaboration Diagram of email client system

A collaboration diagram also known as communication diagram illustrates the interactions between objects work together to achieve a specific goal.

Example collaboration diagram for sending an email

- User
- Email composer
- Email sender
- SMTP server

### Message

User sends a message to email composer to create a new mail.  
Email composer prompts user for recipient, subject and body.  
User provides the necessary information to email composer.  
Email composer creates an email object and sends it to email sender.

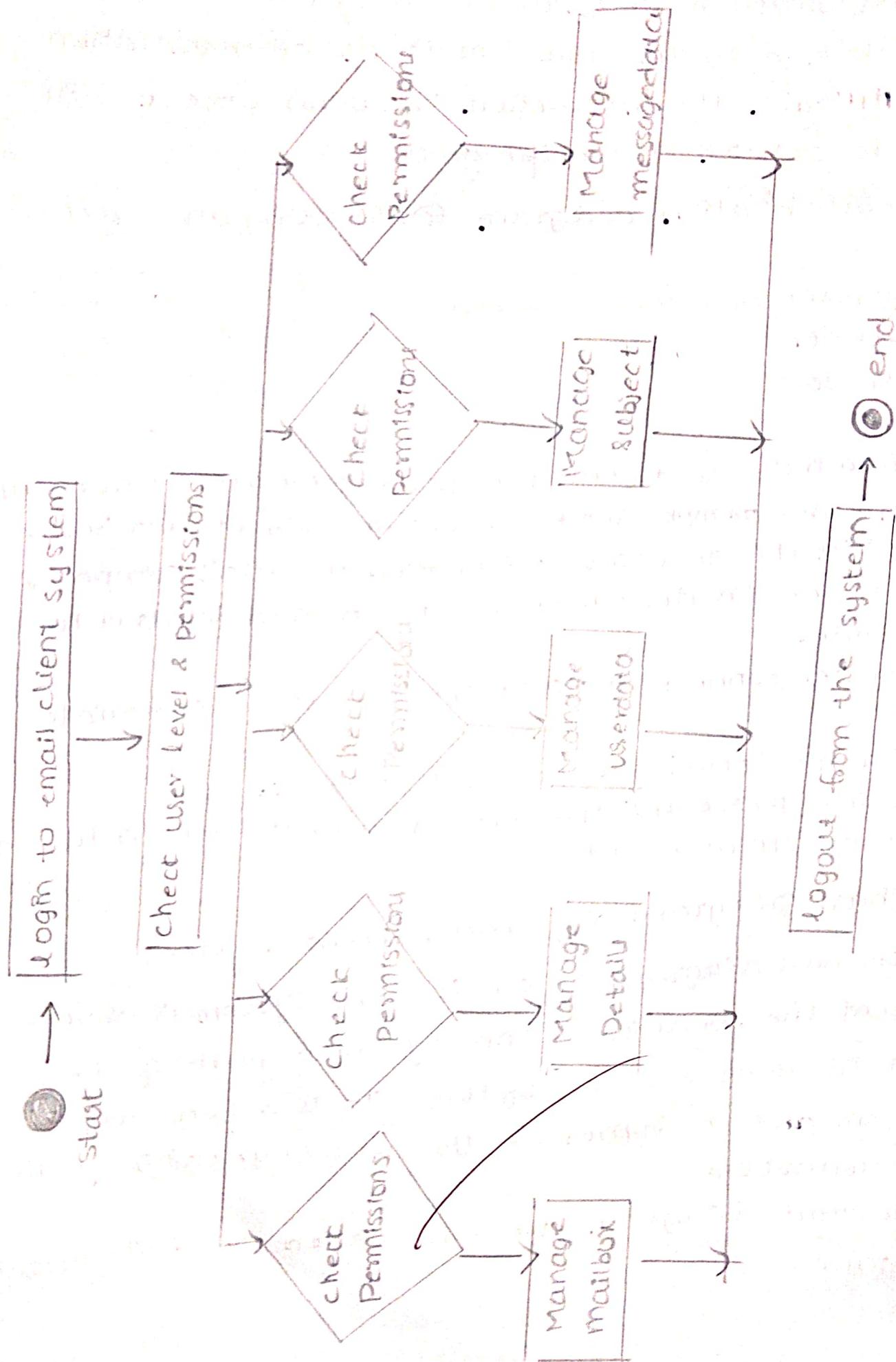
Email sender connects to SMTP server using appropriate protocol.

Email sender sends the email to SMTP server.  
SMTP server processes the email and delivers it to the recipient's email server.

## Statechart Diagram of email client system

A statechart diagram of email client system is used to represent the condition of the system & part of the system at finite instances of time. It is a behavioural diagram and it represents the behaviour using finite state transitions.

Statechart diagram are also called as state machine diagram



## Activity diagram of email client system

The activity diagram of mailing system which shows the flow between the activity of mailbox, messagedata, userdata, mailserviceAddress, Domain. The main activity involved in this UML activity diagram of mailing system are

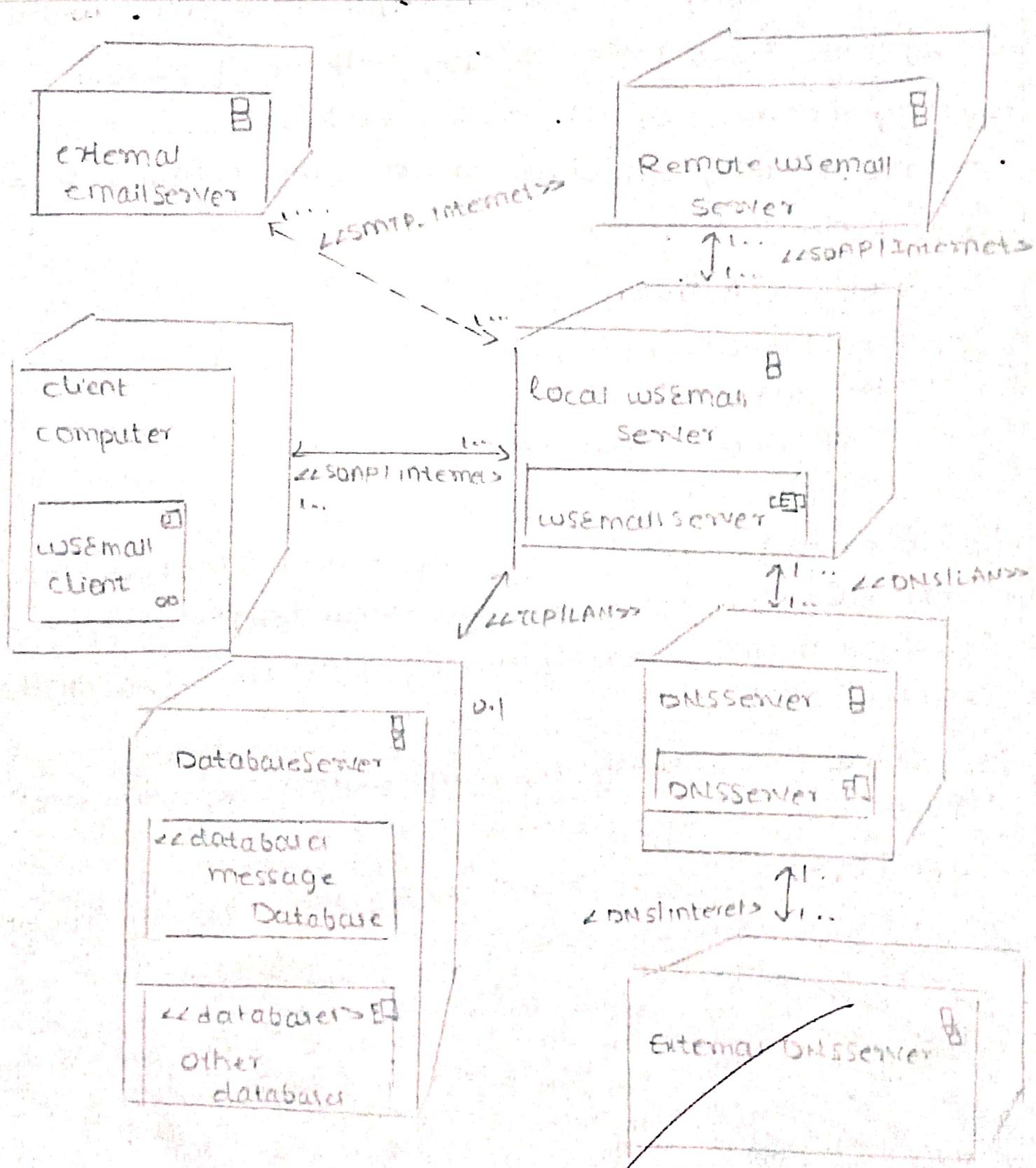
- Mailbox Activity
- Messagedata Activity
- Userdata Activity
- Mailservice Address Activity
- Domain Activity

## Features of activity diagram of email client system

- Admin user can search mailbox, view description of a selected mailbox, add mailbox, update mailbox and delete mailbox.
- It shows the activity flow of editing, adding and updating of message data
- User will be able to search and generate report of userdata, mailservice address, domain

H

## Deployment Diagram for email client System



## Deployment Diagram for email client system

A deployment diagram of email client system is a type of structural UML diagram that shows the physical deployment of software components on hardware nodes. It illustrates the mapping software components onto the physical resources of a system, such as servers, processor, storage devices and network infrastructure. Key elements are nodes, components, artifacts, dependencies, associations, Deployment Specification, communication paths.

### Nodes in Email client system

#### 1. User Device (client)

This could be a laptop, phone or desktop where the email client is running.

#### 2. Email client Application:

The software application used by the user to send, receive and manage emails.

#### 3. Email Server (SMTP / IMAP / POP3)

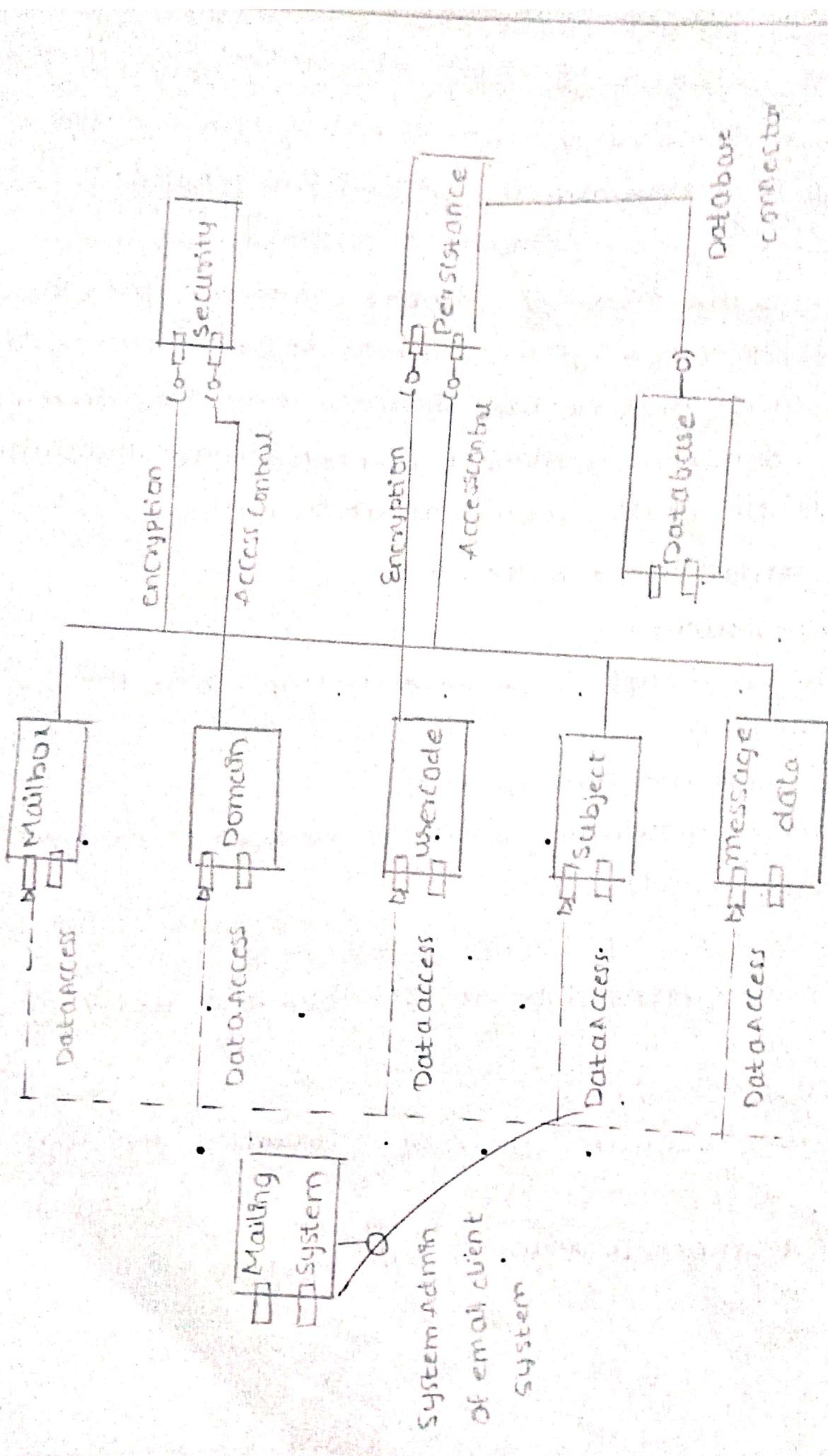
A mail server responsible for sending and receiving emails.

#### 4. Database Server

Used to store email data, user information and logs.

#### 5. Spam Filter

For filtering out spam & junk emails.



## Component Diagram of email client system

## Component diagram of email client system

This is a email client system of component diagram which shows components, provided and required interfaces, ports and relationships between the message data, user data, domain, mailserver address and subject.

This type of diagrams is used in component based development to describe systems with service oriented architecture email client system

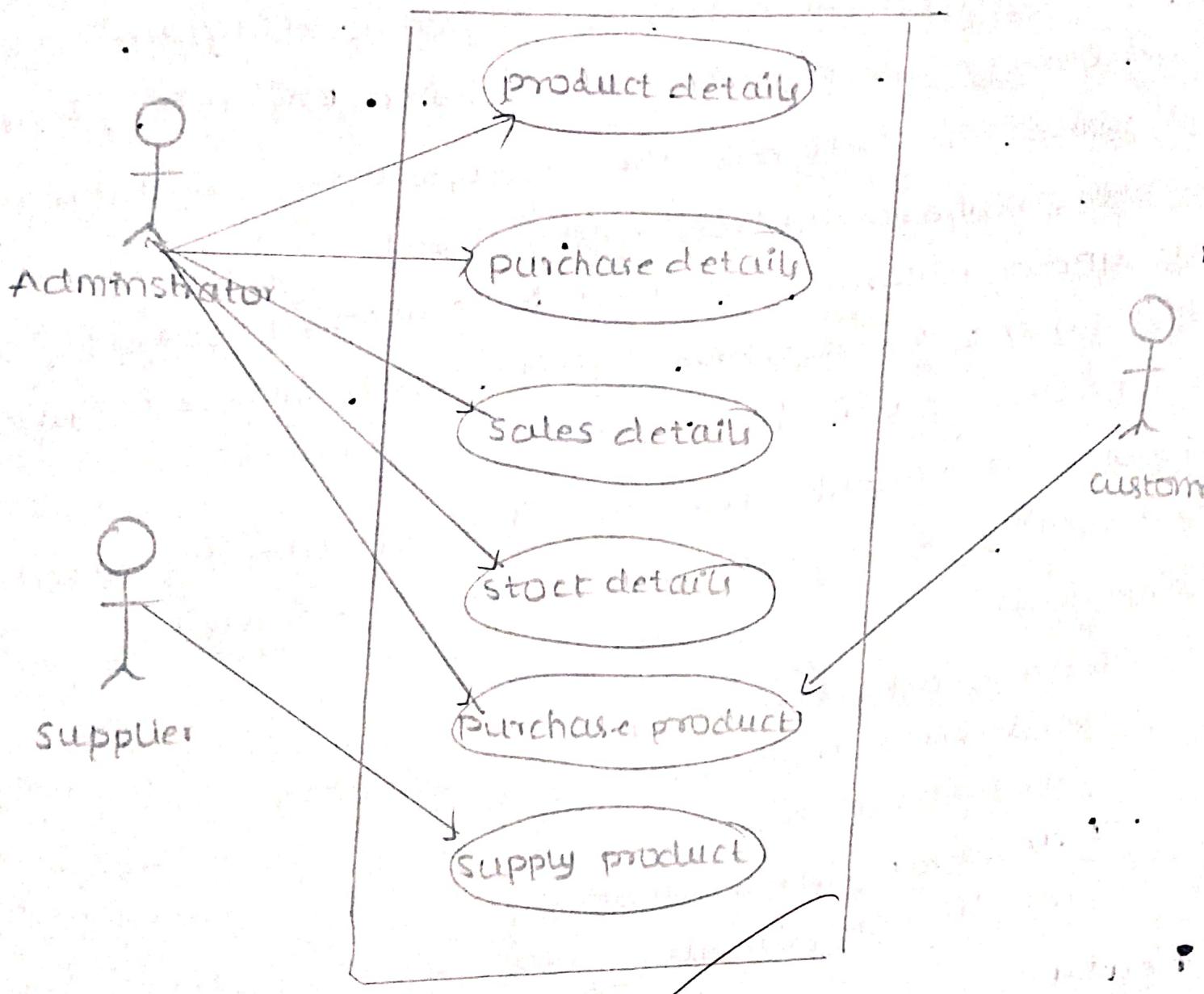
The UML component diagram describes the organization and wiring of physical components in a system

- Message data component
- User data component
- Domain component
- Mailserver Address component
- Subject component

## Features

- You can show the models the components of mailing system
- Model the database schema of mailing system
- Model the executable of an application of mailing system
- Model the systemic source code of mailing system

## Stock maintenance System



## Experiment-4

Aim: Create a complete UML model for stock maintenance system.

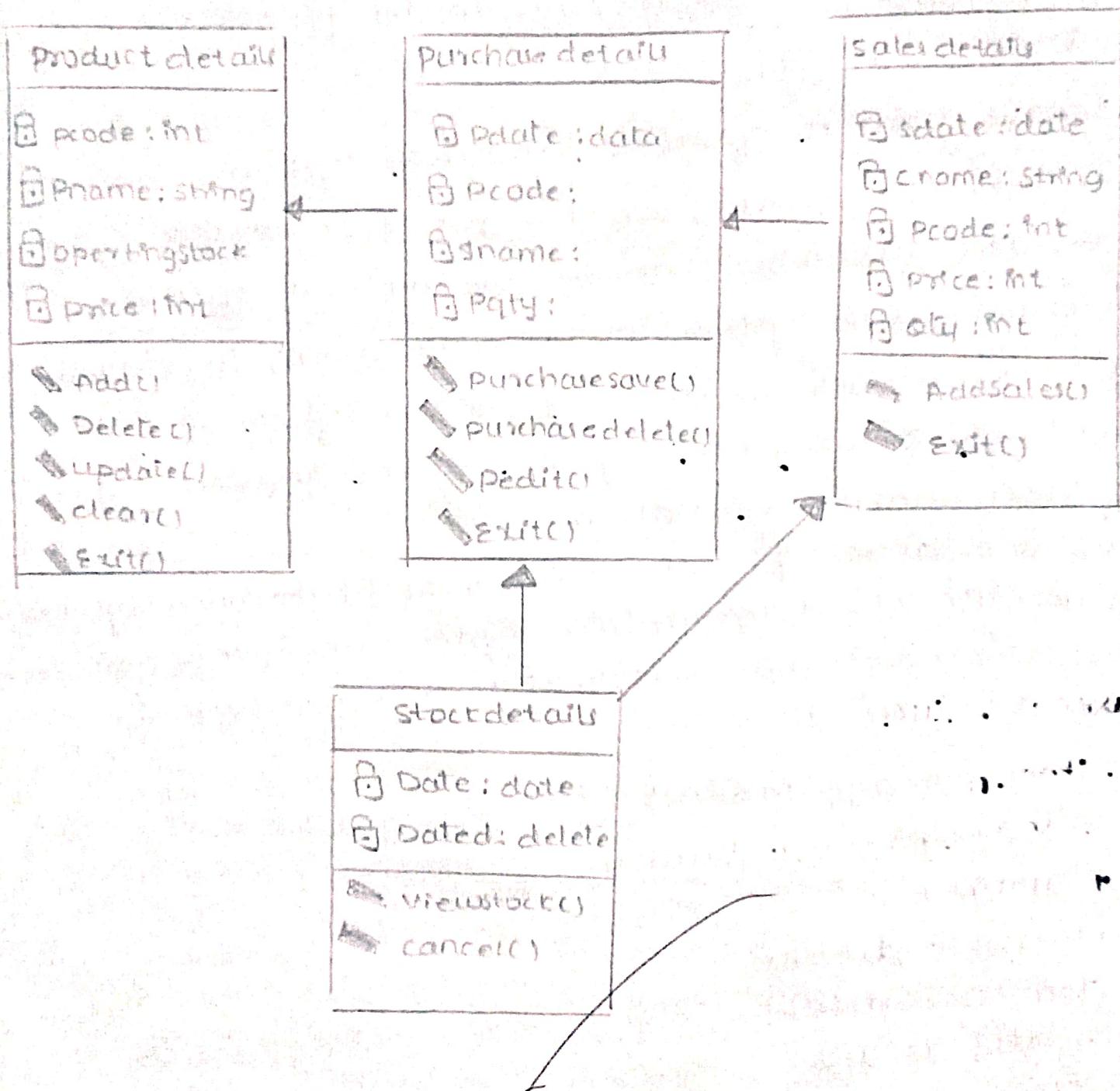
Usecase diagram of stock maintenance system

This usecase diagram is a graphic depiction of the interface interactions among the elements of the stock maintenance system. It represents among the elements of stock maintenance methodology used in system analysis to identify & clarify and organize system requirements of stock maintenance system.

The main actors of stock management / maintenance system in this usecase diagram are system userAgents, customers, and SuperAdmin who perform the different type of usecases such as

- Stock manage product
- Manage product quality
- Manage bill
- Product details
- Purchase details
- Sale details
- Stock details
- Purchase the product
- Supply the product.

# class Diagram of Stock maintenance system



## class Diagram of stock maintenance system

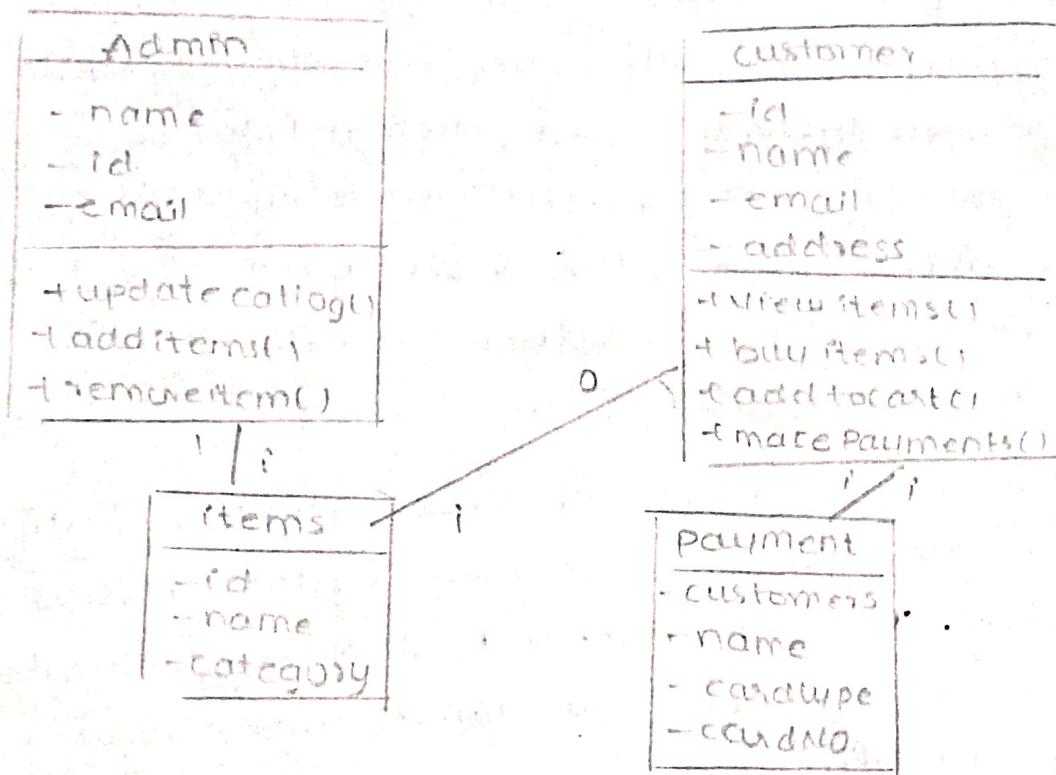
A class diagram in UML describes the structure of stock management system classes, their attributes, operations or methods and the relationships among objects. The main classes of the stock maintenance system are Stock, Product, Product Quality, Bill, Customer, etc.

A class diagram is the type of static structure diagram that describes the system by showing system's classes, their attributes. It is represented using a rectangle with three compartments. Top compartment have class name, middle compartment the attributes and the bottom compartment with operations.

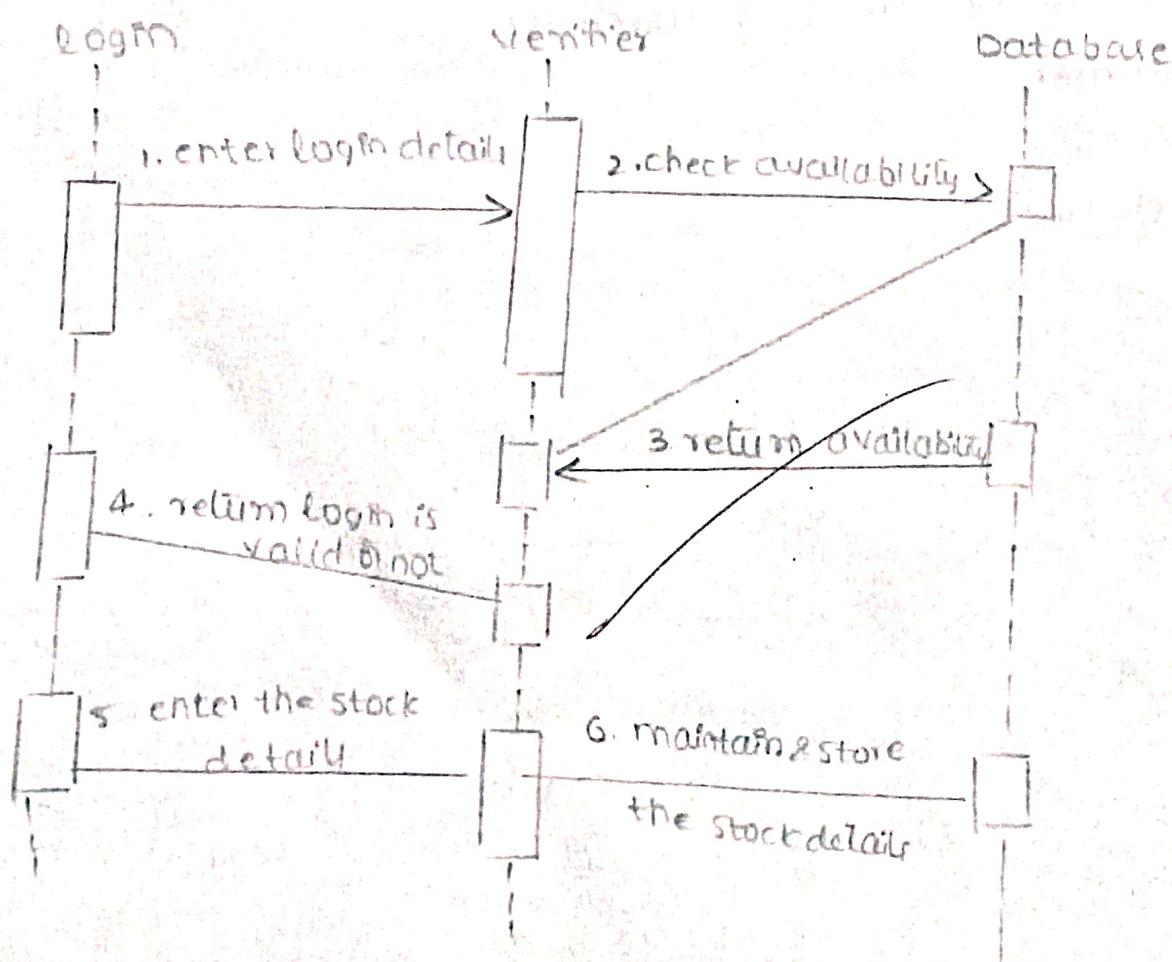
The class diagram has three classes

- i) Customer
- ii) Vendor
- iii) Database

## Object Diagram of stock maintenance system



## Sequence Diagram of stock maintenance system



## Object diagram of stock maintenance system

An object diagram in UML provides a snapshot of the system at a specific point in time, showing instances of classes (objects) and their relationships.

In a stock maintenance system, the object diagram will focus on the key objects such as products, stock, users and transitions, illustrating how they interact at runtime.

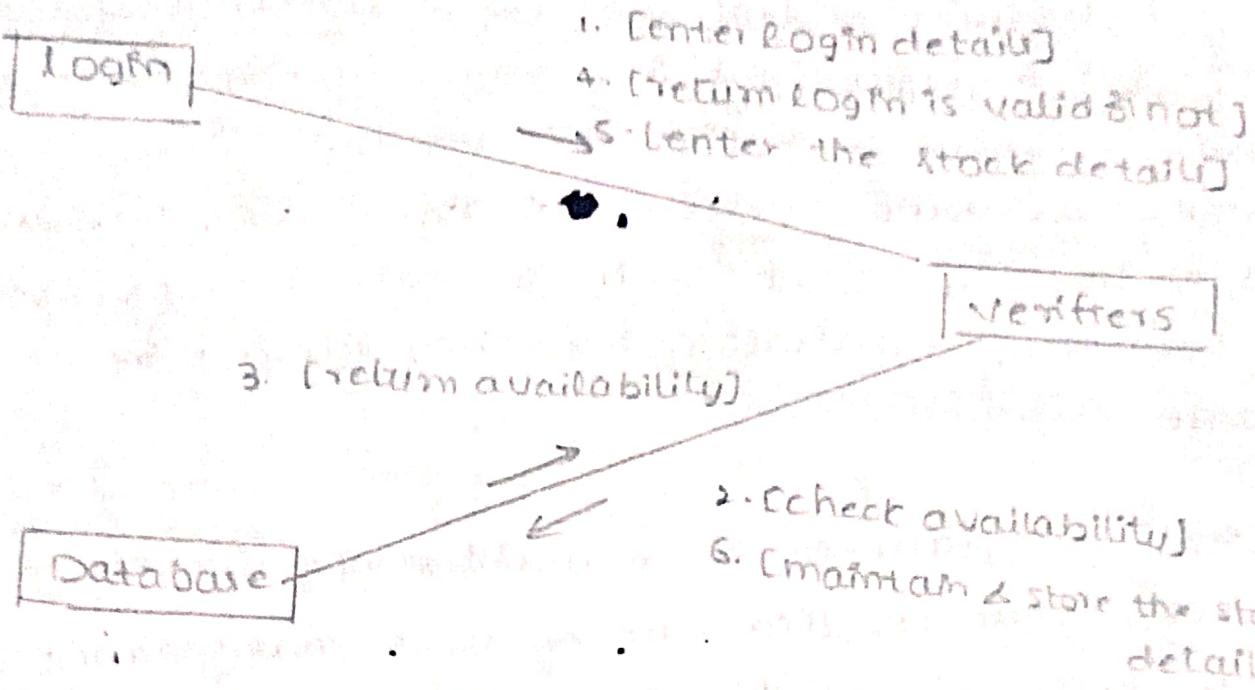
Sequence diagram of stock maintenance system  
This UML sequence diagram of stock maintenance system which shows the interaction between the objects of stock, product, customer, store, quality. The instance of class objects involved in the UML sequence diagram of stock management system.

A sequence diagram in UML is a kind of interaction diagram that shows how processes operate with one another and in what order. It is construct of a message sequence chart.

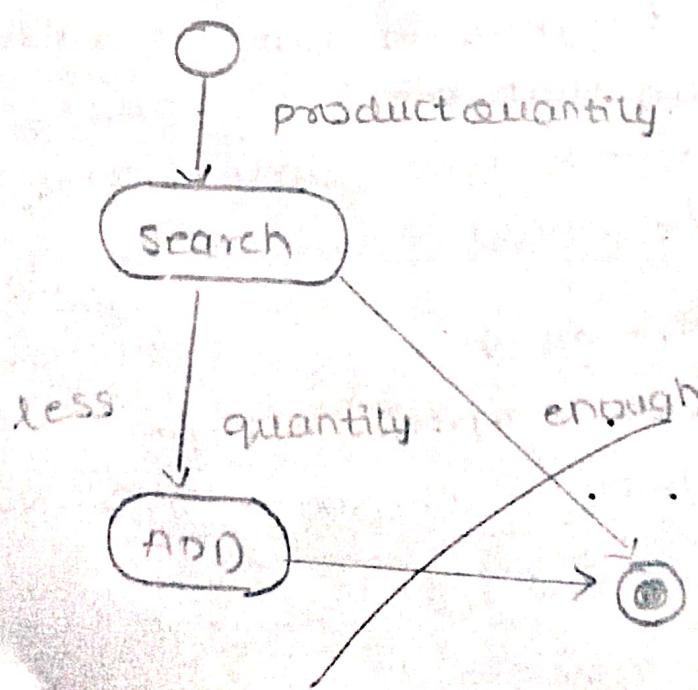
There are two dimensions

- 1- Vertical dimension - represent time
- 2- Horizontal dimension - represent different ...

## Collaboration diagram Stock maintenance



## Statechart diagram of Stock maintenance system



## Collaboration diagram of stock maintenance system

A collaboration diagram also called a communication diagram or interaction diagram. A sophisticated modelling tool can easily convert the collaboration diagram and the vice versa. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as overall operation of the system in realtime. The collaboration diagram is to show how the customer login and places the order in the system.

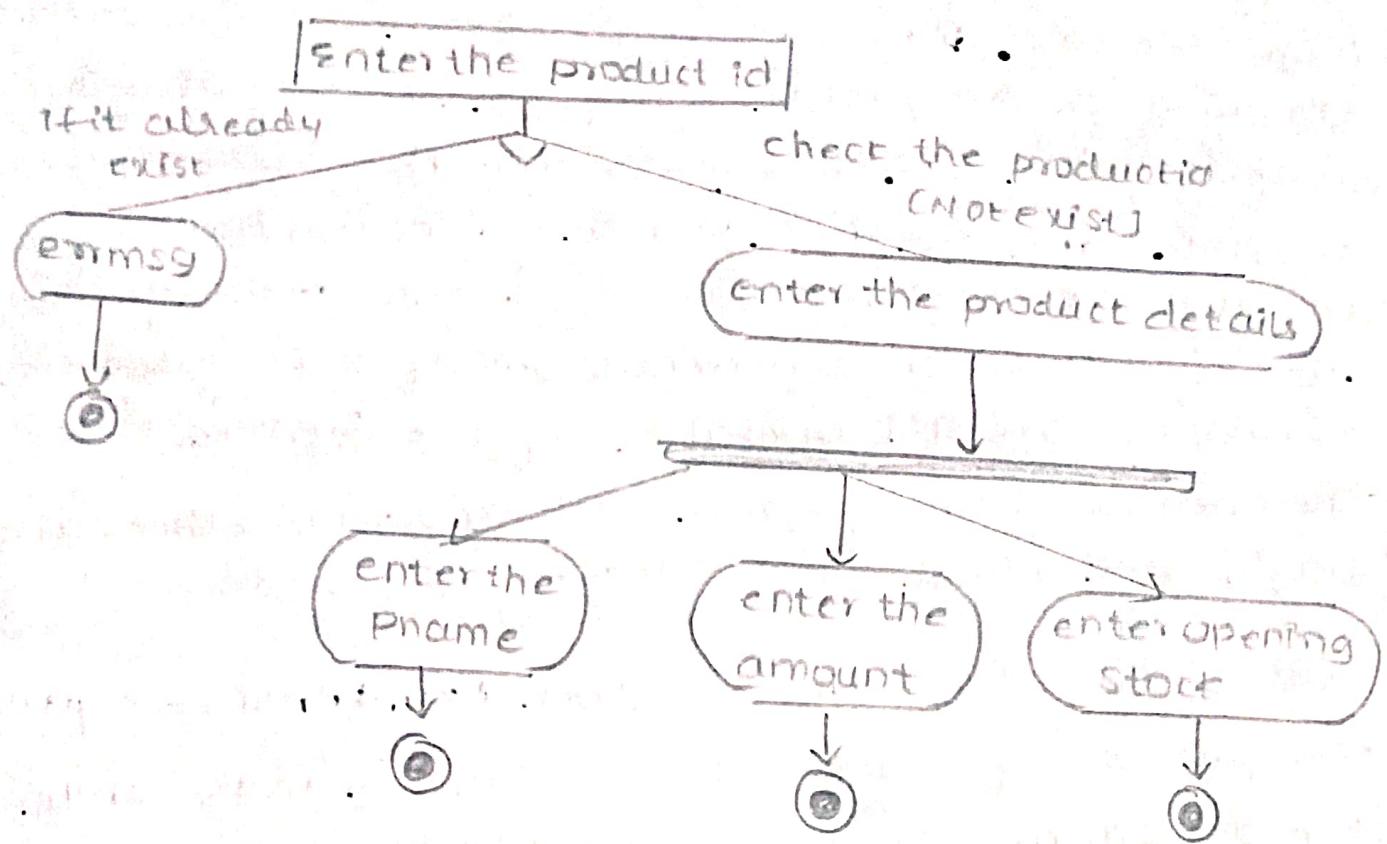
## Statechart Diagram of Stock Maintenance System

The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. A final state is represented as circle surrounding a small dot.

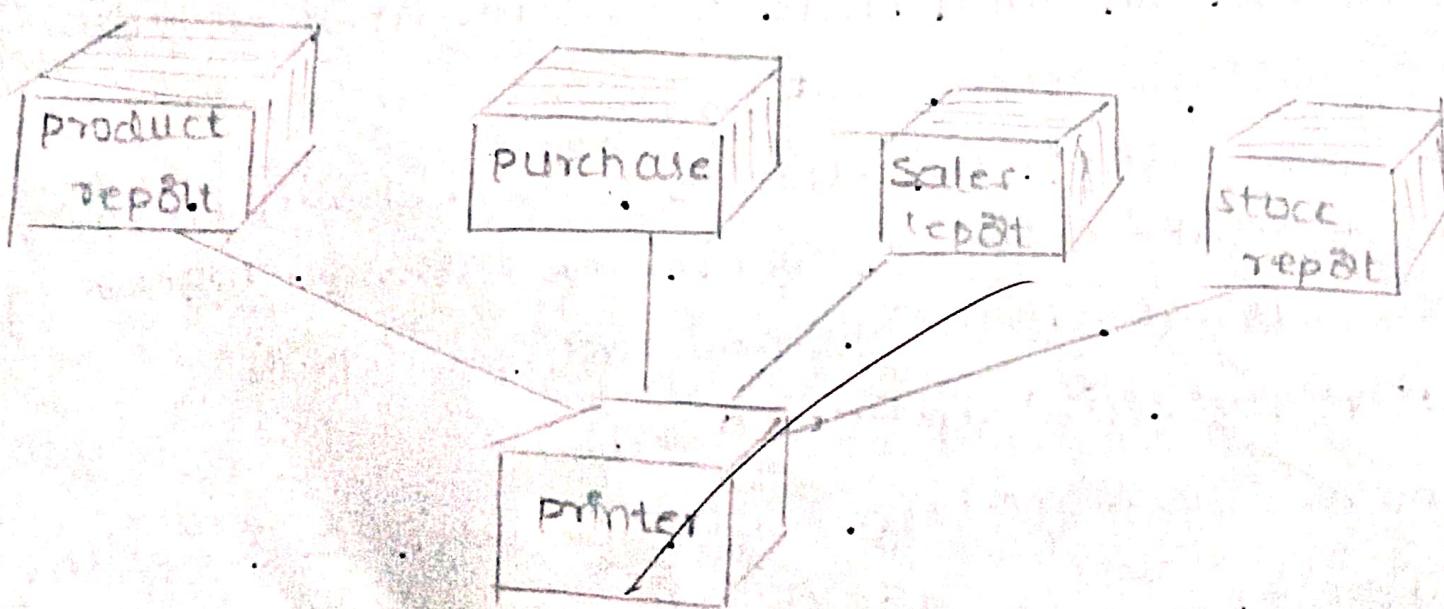
- The state diagram describes the behaviour of system.
- First state is Login where the customer login to system.
  - The next state is to fill the customer details.
  - Next state is to place orders.

~~update the database with the orders and details of customer.~~

## Activity diagram of stock maintenance system



## Deployment diagram of stock maintenance system



## Activity Diagram of stock maintenance system

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the UML activity diagram can be used to describe the business and operational step by step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as an rounded box containing name of operation. The stepwise activities performed in stock maintenance system.

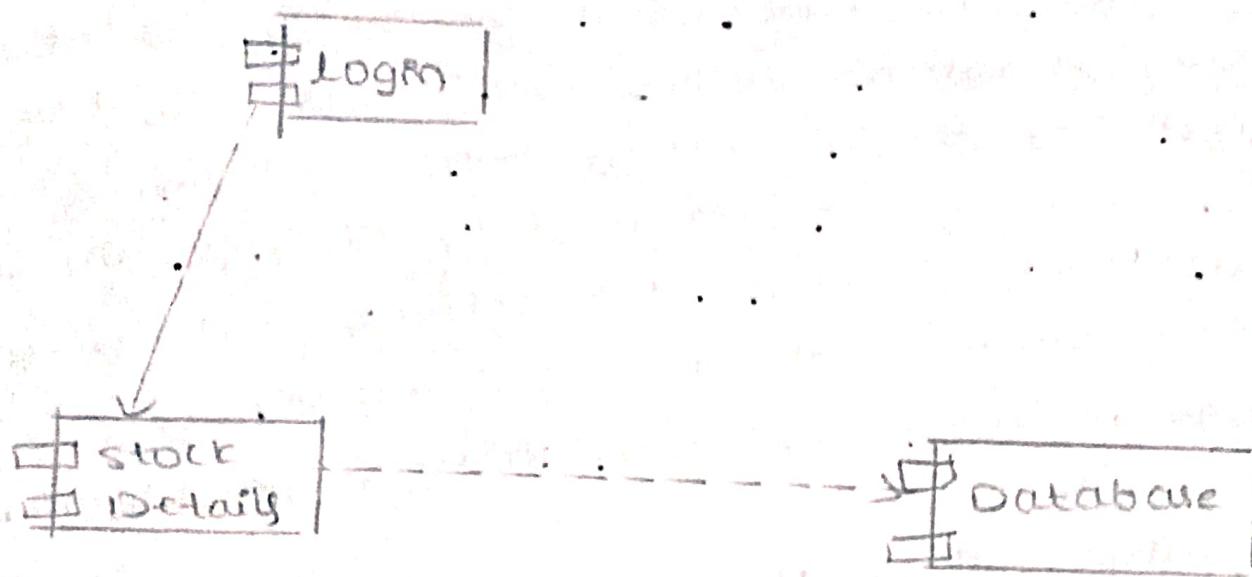
First the customer login then fill the details

- a) The customer places the Order according to their need
- b) After placing Order the database is updated
- c) Vendor login to system and verifies the customer Order and stock details

## Deployment Diagram of stock maintenance system

A deployment diagram in UML shows the physical arrangement of hardware and software in the system for a stock maintenance system. The deployment diagram will show how different components interact across various physical locations, such as the client machine, application server and database server.

## Component Diagram of stock maintenance system



## Component Diagram for stock maintenance system

This component diagram main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association.

This is a component diagram of stock management system which shows components provided and required interfaces, parts and relationships between the store, Bill, product, stock and quality. This type of diagram is used in component-based development to describe systems with service-oriented architecture. Stock maintenance system UML component diagram, describes the organisation and working of the physical components in a system.

## Experiment-5

**Aim:** Consider the user's view of your respective system  
 Identify the usecases, actors involved in the system and  
 develop the usecase and subusecase diagram.

An android application to address the various needs within the college community. The app aims to streamline access to course materials, provide an interactive campus map, showcase campus events and reminders, facilitate academic progress tracking, offer student support services, ensure campus safety, foster social networking and simplify course registration and schedule management.

The aim is to enhance the overall college experience for students, faculty and staff.

### Usecase diagram

It is a graphic depiction of the interaction among the system. It represents the methodology used in system analysis to identify, clarify and organise system requirements.

#### Actors

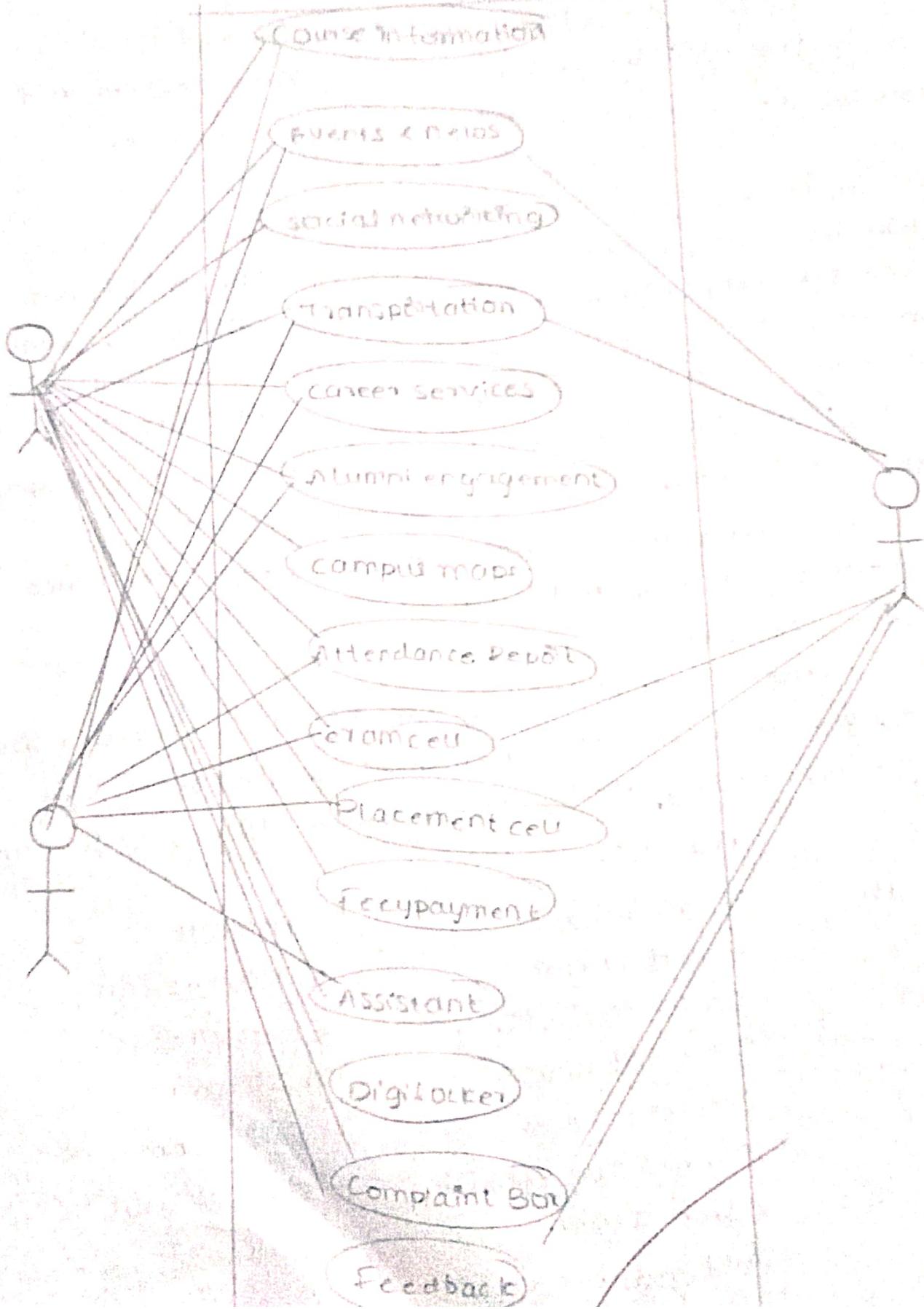
Students  
faculty  
Admin

#### usecases

Courseinfo  
Events & news  
Social networking  
Transportation  
Career services  
Alumni engagement  
Campus maps  
Attendance report

Examcell  
Placement cell  
Fee payment  
Assistant  
Digilocker  
Complaint box  
Feedback

## Campus Information



usecase diagram.

Usecase diagram describes the functionality of a system and users of the system.

Usecase diagram are considered for high level requirement analysis of a system. so when the requirements of a system are analyzed the functionalities are captured in usecases.

So we can say that the usecases are nothing but the system functionalities written in an organized manner.

Things which are relevant to usecases are actors. Actors are defined as something that interacts with the system.

Functionalities to be represented as an usecase  
Relationships among the usecases and actors.

→ Usecase diagrams are drawn to capture the functional requirements of a system. so after identifying the above items we have to follow the guidelines

→ The name of a usecase is very important so that it can identify the functionalities performed.

→ Give a suitable name for others

→ Show relationships and dependencies clearly in diagram

→ Do not include all types of relationships Because the main purpose of the diagram is to identify the requirement

→ We note whenever required to clarify some of the important points.



## Subusecase diagram

A subusecase diagram is a more detailed and specific representation of a particular usecase within a larger system. It focuses on a particular functionality or module, breaking down how various actors interact with that specific portion of the system. The diagram provides a visual overview of user interactions and the processes involved in achieving a certain goal, offering more clarity than a general usecase diagram for those focused areas.

### 1. Definition and purpose

A subusecase diagram is essentially a more refined version of usecase diagram tailored to highlight a specific part of the overall system. Its purpose is to ensure that both technical and non-technical stakeholders can easily comprehend how individual components or features of a system function, specifically focusing on user action and system responses.

### 2 Components of Sub-usecase Diagram

Actors: These are the people or systems that interact with the part of the system being modeled.

Usecases: These are the actions or functionalities that the actor can perform within the scope of the sub-system. Each usecase typically has a name that reflects the action being performed.

**Relationships:** There are lines connecting actors to usecases showing the interactions between them. Relationships can also exist between usecases themselves, indicating dependencies or extensions.

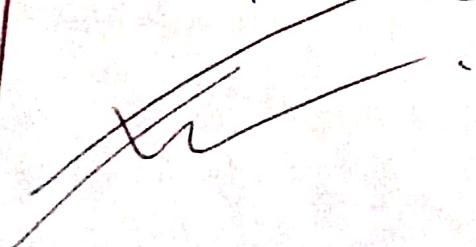
### Advantages

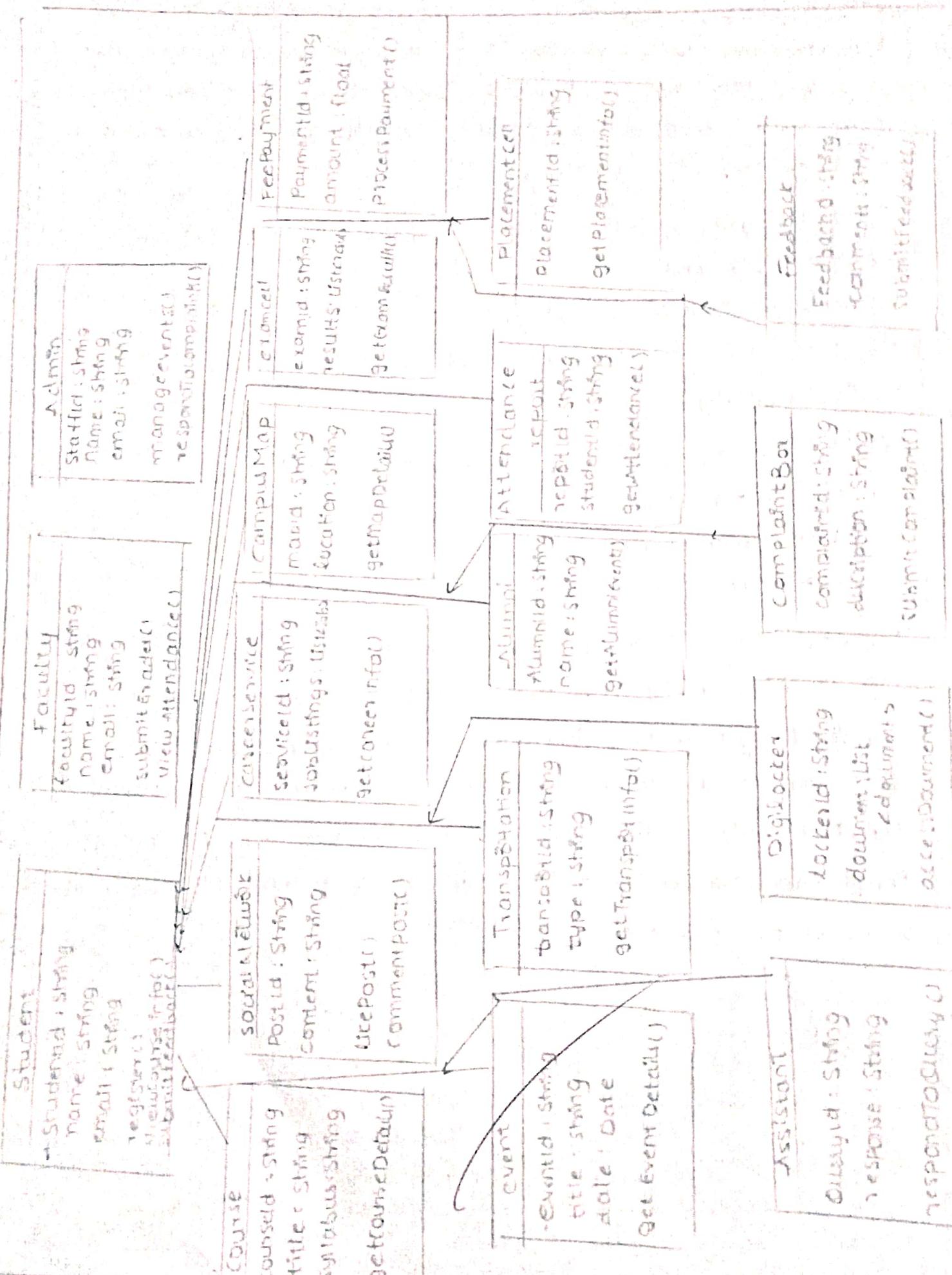
- Clarity & Focus
- Improved communication
- Supporting development and Testing
- Modularity

**Challenges in creating subusecase diagram**

- Over complication
- Interdependences
- Scalability

Incorporating subusecase diagrams into our workflow can significantly enhance clarity and ensure that detailed system requirements are met accurately. This approach ensures that every aspect of systems functionality is thoroughly understood and properly implemented, reducing the risk of miscommunication or incomplete development.





## Experiment - 6

Aim: Consider the structural view of your respective system.

- Identify the classes, their attributes, methods, relationships and develop the class diagram.
- Identify the objects and their links between and develop the object diagram.

### Class diagram

A class diagram is a type of static structure diagram in UML that shows the structure of a system by representing its classes, attributes, methods and the relationships among objects.

Key elements:

Class: Represented as a rectangle divided into 3 sections

Top: class name, middle: Attributes, Bottom: Methods

Attributes: Characteristics or properties of the class

Methods: Functions or operations that can be performed by class

Eg: class : student

Eg: Placement class.

Attributes: Company ID, Company Name, JobPosition

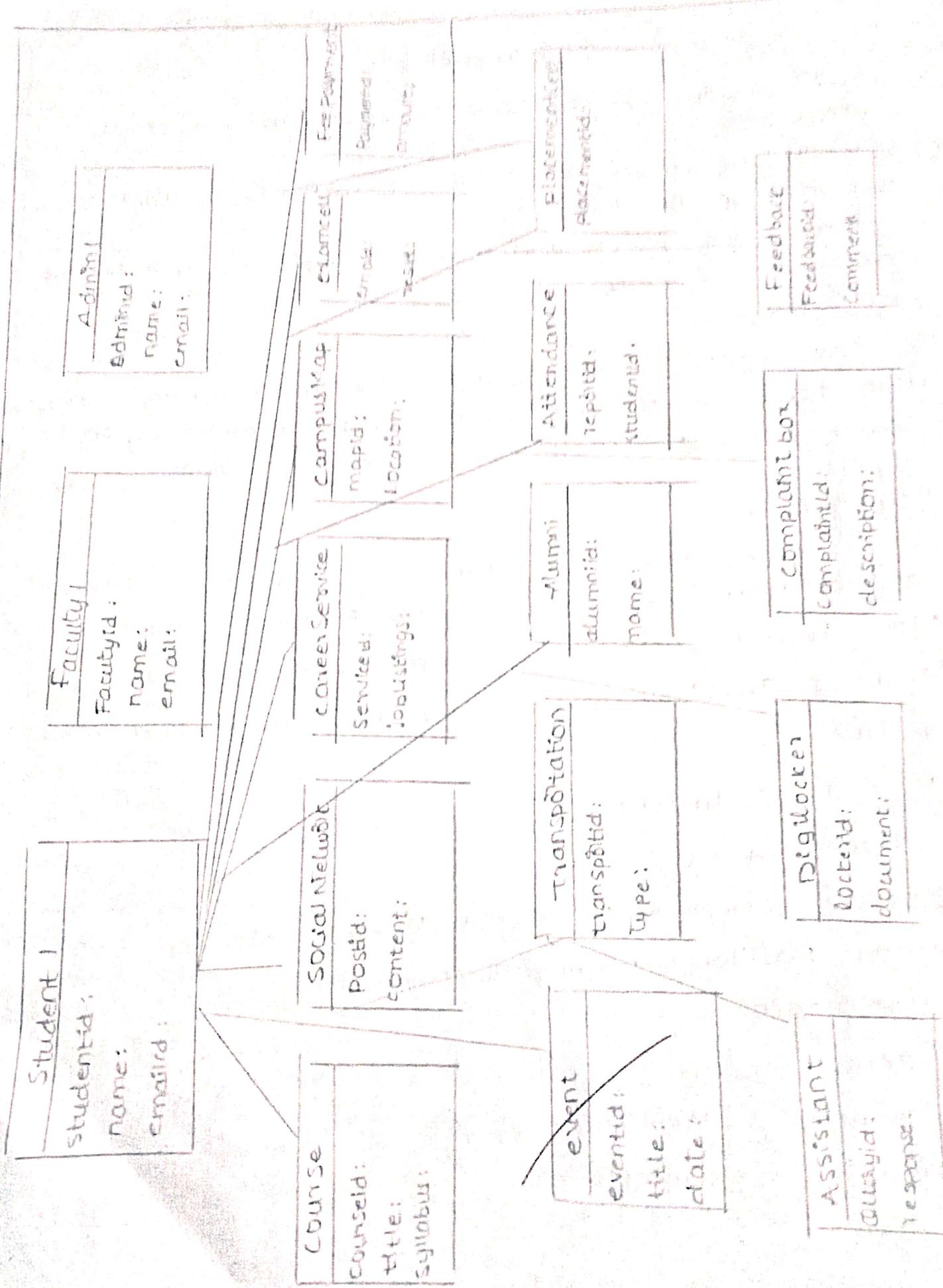
Method: PostJob(), managePlacement()

Relationships

Student → course

Faculty → attendance report

Admin → Examcell.



## Object Diagram:

An object diagram is a type of structural diagram in UML that represents a snapshot of instances in a system at a specific moment in time, while class diagram focus on the blueprint or general structure of the system by showing classes, their attributes and their methods.

Object diagrams zoom in on the actual objects and their state at a particular point. This diagram is especially useful for visualizing the real world scenario of a system's behaviour and structure during execution, rather than in the abstract.

Key elements:

Objects: Objects contain specific attribute values rather than attribute definition.

Links: Links in an object diagram can represent association, aggregations depending on the nature of relationship between the instances.

Attributes & values: Attributes in object diagram are shown with their specific values at a point in time.

Multiplicity: Multiplicity of relationships between objects is often displayed.

States: Object diagram may also depict the state of the object depending on the context.

### Advantages

- Concrete Representation
- Clarity in relationships
- Simplification of complex system
- Documentation and analysis.

O

Counseling Events

Request course info  
Provide course details  
Request event info

Provide event list

Join clubs

Information

Request transport info

Provide transport info

Request job listings

Provide job listings

Request Alumni events

Provide Alumni events

Request campus maps

Provide navigation tools

Request Attendance report

Provide attendance report

Request exam results

Provide placement assistance

Request placement info

Request fee payment

Confirm payment

Ask doubts

Provide assistance / response

Access Digiblocker

Provide info

Request list item

Get confirmation of request

Social media	News	Career services	Campus maps	Attendance report	Placement	Fees	Fee Payment	Complaints	Feedback
Request course info	Provide course details	Request event info	Provide event list	Join clubs	Information	Request transport info	Provide transport info	Request job listings	Provide job listings

Feedback	Fee Payment	Complaints	Attendance report	Placement	Campus maps	News	Career services	Social media	Events
Request list item	Get confirmation of request	Provide assistance / response	Access Digiblocker	Provide info	Request list item	Request fee payment	Confirm payment	Ask doubts	Request course info

## Empirical

- Aim consider the Behavioural view of your respective system.  
visualise and justify runtime framework of the system and develop the sequence diagram by using life-line message execution, occurrence, interaction fragments.  
Develop the communication diagram to portray the object architecture in the system.

### Sequence Diagram

A sequence diagram is a type of interaction diagram in UML that depicts how objects or components in a system interact with each other over time. It represents the flow of messages between objects in order they occur.

### Components of Sequence diagram

- 1-Actors
- 2- Objects
- 3 - Lifelines
- 4- Activation Bars
5. Messages
  - synchronous messages
  - asynchronous messages
  - Return messages

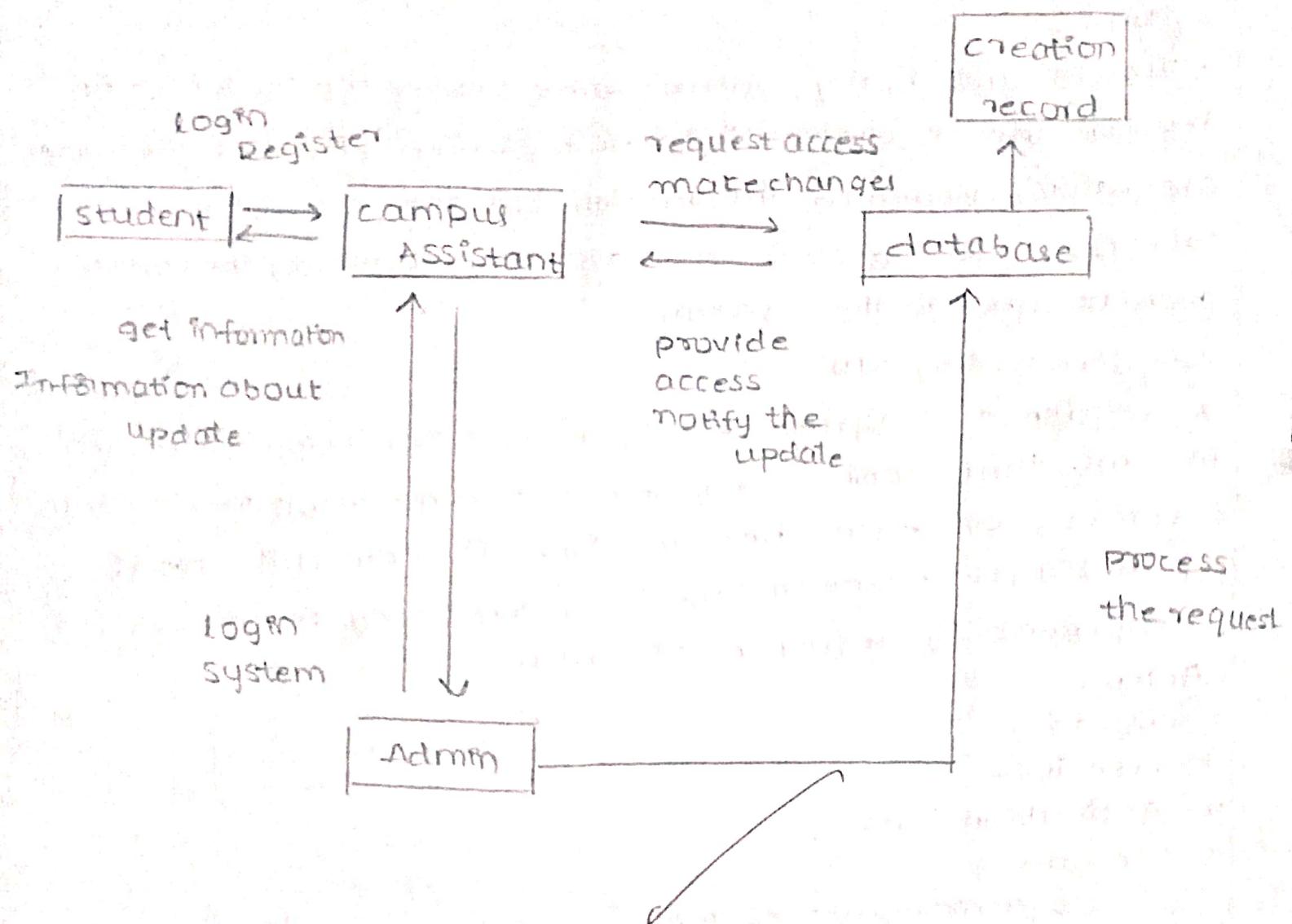
- 6 Gates

- 7 Fragments.

### Benefits of sequence Diagram

- 1-clear visualisation of interactions
- 2-Time Based Representation
- 3-Easy to Flow logic
- 4-Encourages Better system design
- 5-Supporting documentation

## Collaboration diagram for campus assistant



## Collaboration diagram

A collaboration diagram also known as a communication diagram in UML is a type of interaction diagram that focuses on how objects or components interact to fulfill a particular task & process. Unlike sequence diagrams which emphasizes the time ordering of messages, collaboration diagrams emphasize the structural relationships between objects. The main goal of a collaboration diagram is to show the organization of objects and the links between them, alongside the messages exchanged.

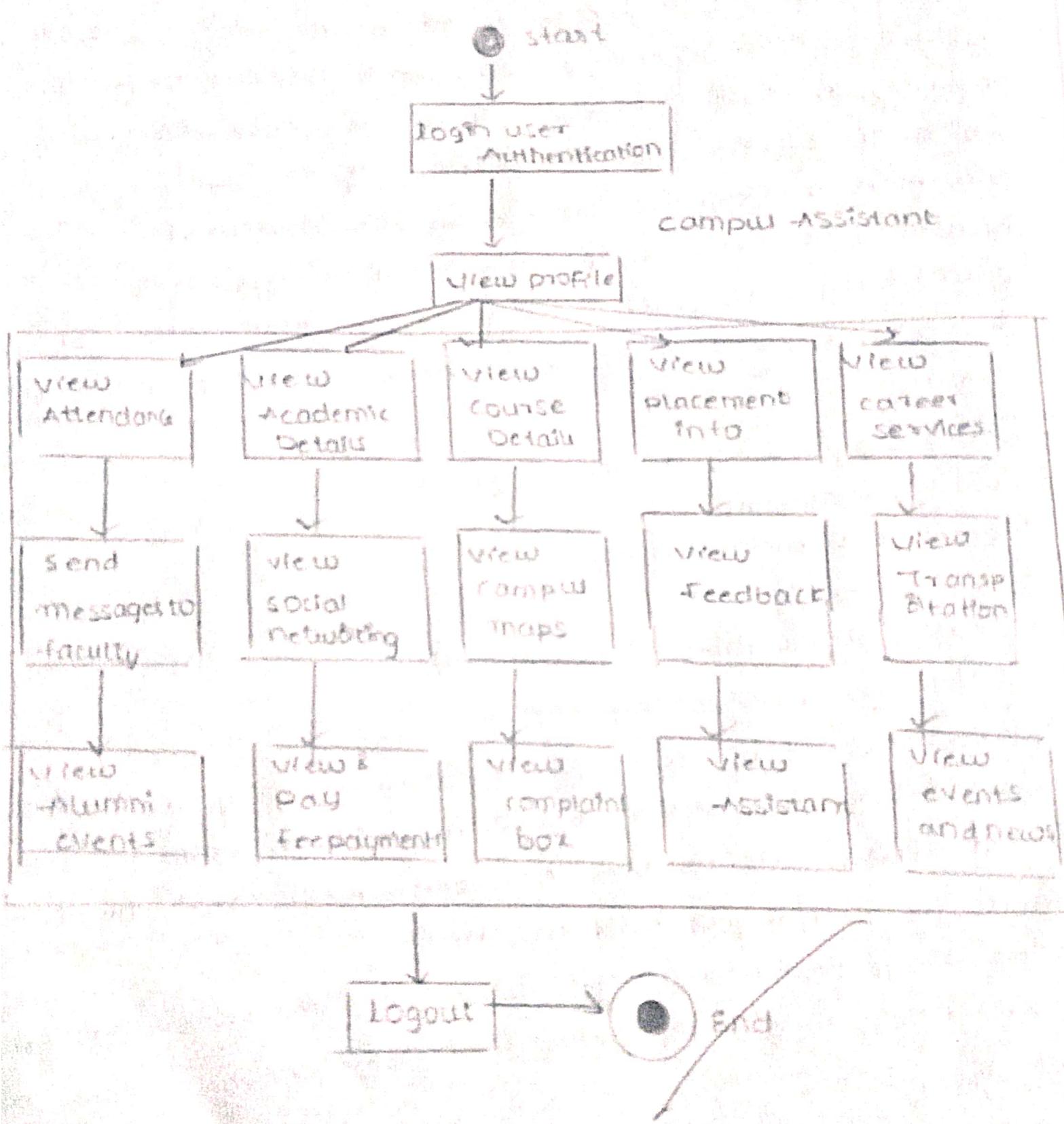
### Benefits of collaboration diagram

- Focus on structure
- compact representation
- object centric view
- clear visualization of object roles
- useful for reverse engineering

Collaboration diagrams offer a valuable way to visualize object interactions within a system, with an emphasis on the structural relationships between objects and the messages exchanged between them.

They are especially useful in object-oriented design for representing how components work together to achieve a task

# State chart / State machine diagram of campus-Assistant



## Experiment-8

Aim: Consider the Behavioural view of your respective System. Develop the dynamic view of the system that portrays the behaviour of the system using statechart diagram.

Develop the activity diagram to demonstrate the flow of control within the system by considering concurrent and sequential activities.

### Statechart Diagram

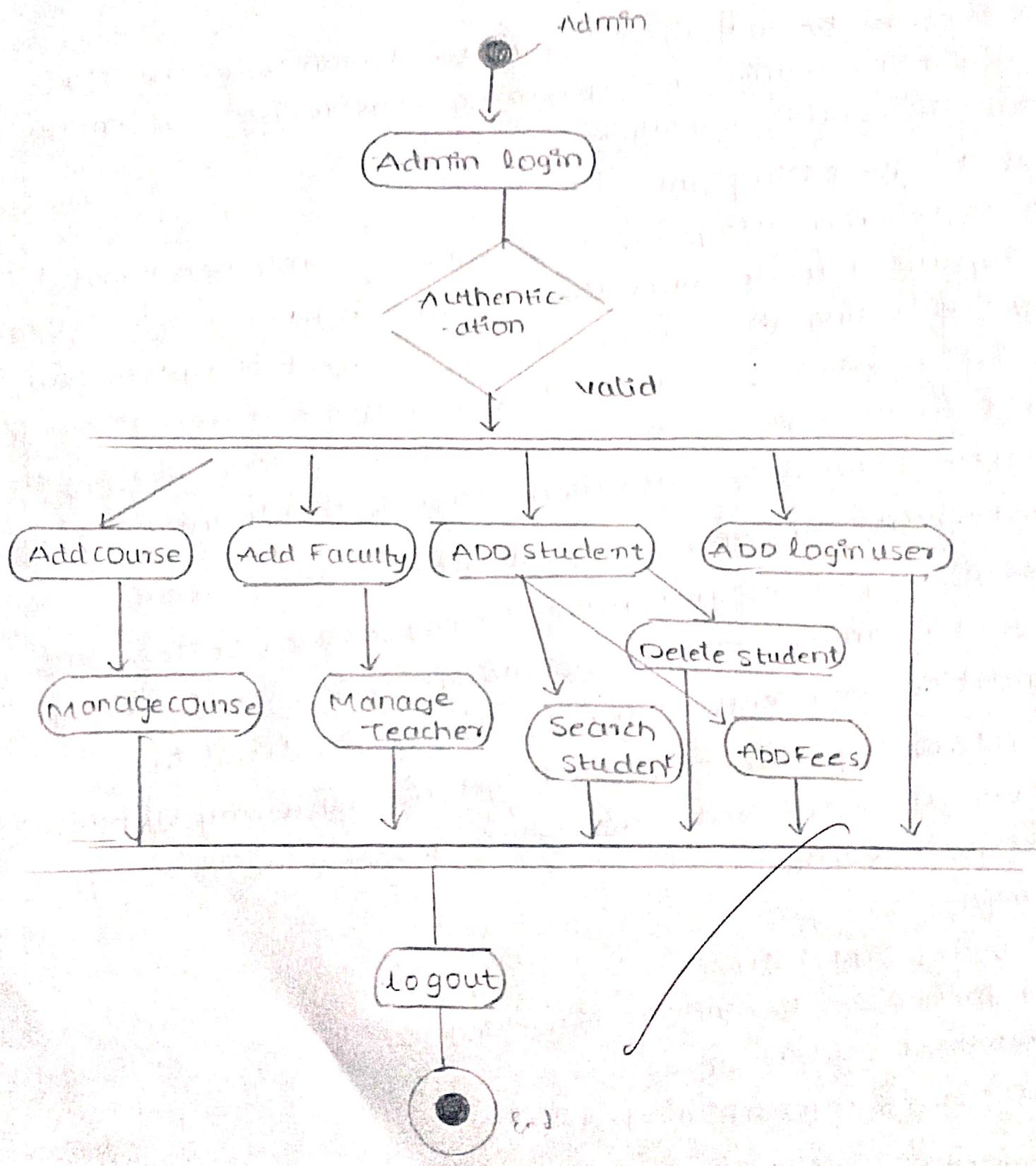
A state chart diagram is a type of UML behavioural diagram that describes the dynamic behaviour of a system by illustrating the various states an object or system can be in as well as transitions between those states. It focuses on how an object responds to different events during its lifetime, providing a detailed view of its internal workings.

Statechart diagrams are particularly useful when designing systems that have clearly defined states and need to handle complex behaviour based on different conditions or events. They are commonly used in realtime systems, embedded systems, and any systems where an object's state changes in response to internal or external events.

### Benefits

- 1- Clarity & Simplicity
- 2- Comprehensive Behaviour Modelling
- 3- Improved system design
- 4- support for hierarchical states.

## Activity diagram of campus Assistant



## Activity Diagram

An Activity Diagram is a type of behavioural diagram in UML that visually represents workflow or stepwise activities and actions, along with the flow of control and data. Activity diagrams are widely used to model the dynamic aspects of systems especially those that focus on processes or workflows. They are similar to flowcharts but provide more functionality by allowing you to represent parallel, conditional and sequential activities.

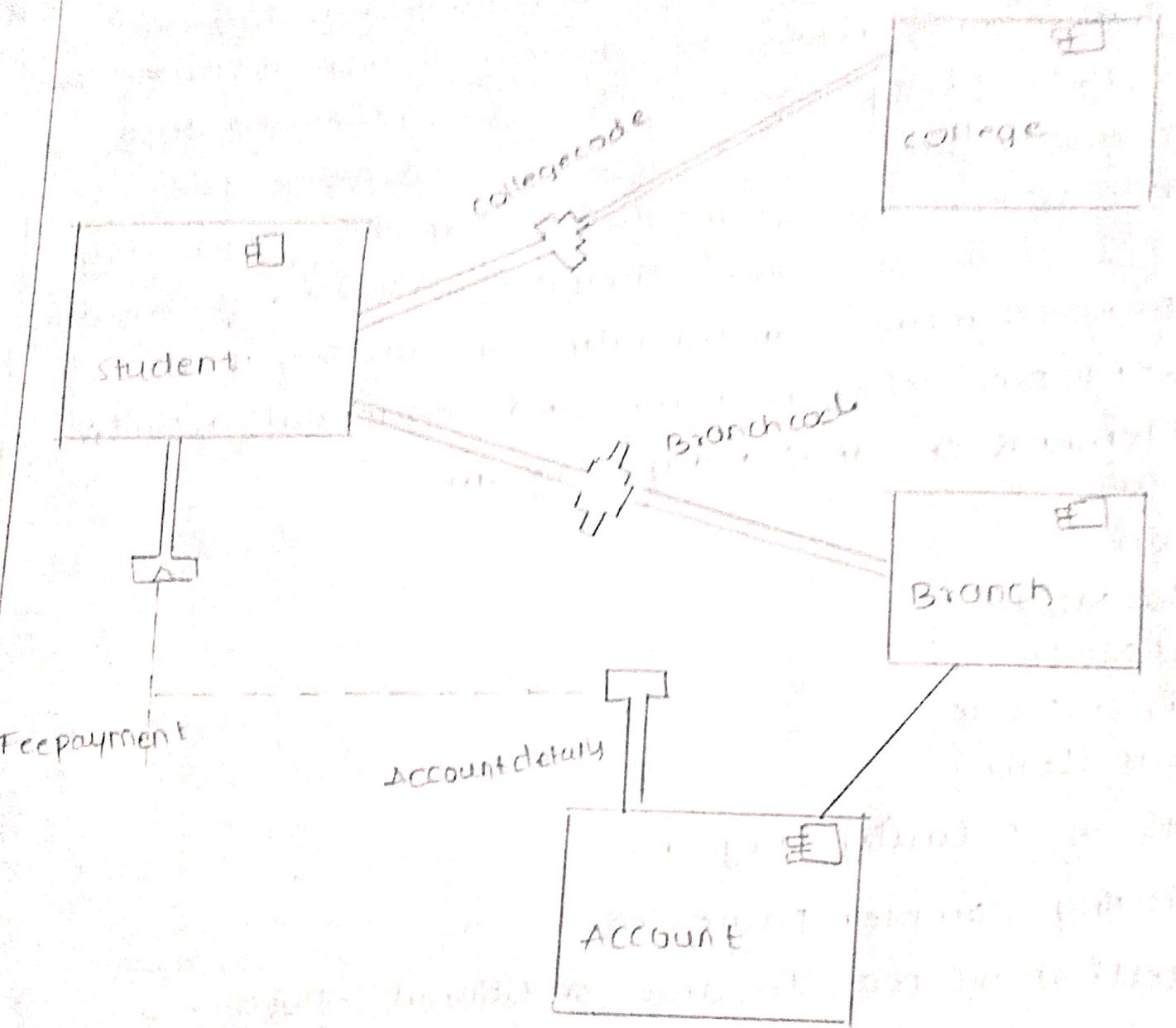
### Key elements of an Activity Diagram

1. Activity
2. Action
3. Initial Node
4. Final Node
5. Decision Node
6. Control flow

### Benefits of Activity diagram

1. Clarifying complex processes
2. Modelling of parallel and conditional logic
3. Improved communication
4. Flexibility
5. Easy integration with other UML diagrams

Activity diagram are an effective tool for visualizing workflow and process in a system, providing a clear and detailed view of how activities, decisions and parallel actions flow.



Component diagram

## Experiment-9

Aim: Consider the Environment implementation view of your respective systems.

Develop the component diagram that visualizes the relationships as well as the organisation between the components present in the system.

### Component Diagram

A UML component diagram is often used to visualize high level architectures in a system, showing how different parts of the system interact and depend on each other.

#### Usage:

Architectural View: Helps in understanding the structure of a system at a higher level without delving into implementation details.

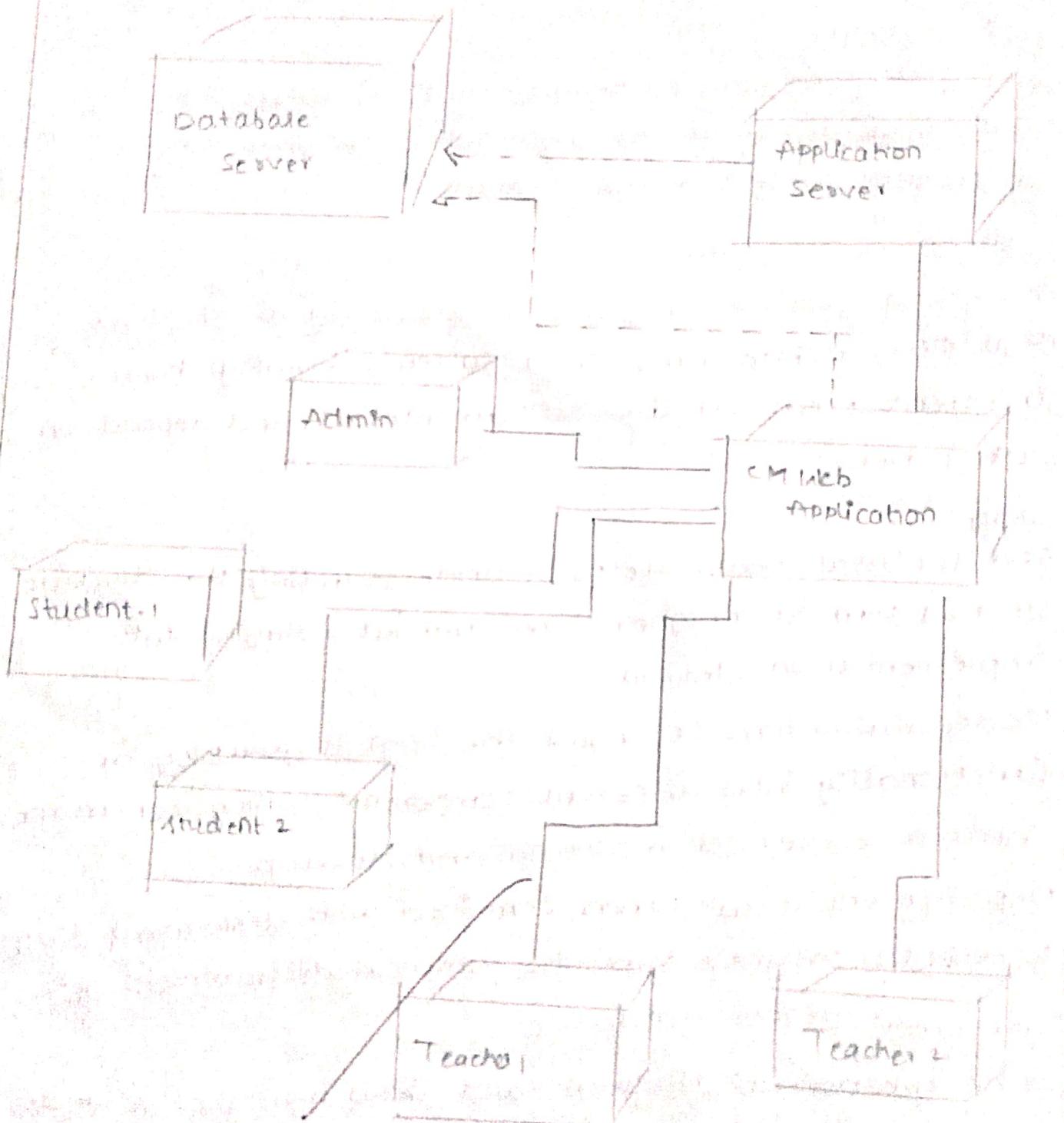
Modularization: It shows the logical grouping of functionality into different components, which can make the system easier to maintain and develop.

Deployment View: When combined with deployment diagram it provides insights into the physical distribution of components across hardware.

The component diagram could show how the UI sends requests to Backend, which in turn depends on the Database for persistent data storage.

Xm

## Deployment diagram



## Experiment-10

Aim: consider the environmental view of your respective system

Develop the deployment diagram to depict how software interacts with hardware to perform its execution by identifying nodes and their relationships in the system

### Deployment diagram

A deployment diagram shows the physical deployment of aircrafts on hardware nodes

The diagram provides a view of system hardware and the relationships between deployed software and the hardware environment.

Deployment diagram are crucial in understanding system scalability, redundancy, and physical configuration for production environment.

### Advantages of deployment diagram

- 1- Physical structure visualisation
- 2- Improved communication
- 3- System scalability
- 4- Performance optimization
- 5- Security planning
- 6- Troubleshooting and maintenance
- 7- Resource Management
- 8- Clarity for different environments.