

# Certificate

Name: Khushi

Class: I TE-2

Roll No: 031

Exam No:

Institution \_\_\_\_\_

*This is certified to be the bonafide work of the student in the  
Laboratory during the academic  
year 2021 / 2022.*

No. of practicals certified \_\_\_\_\_ out of \_\_\_\_\_ in the  
subject of Software Engineering And Project Management

.....  
Teacher In-charge

.....  
Examiner's Signature

.....  
Principal

Date: .....

Institution Rubber Stamp

(N.B: The candidate is expected to retain his/her journal till he/she passes in the subject.)

## EXPERIMENT- 1

AIM → To write a Problem statement

THEORY :

### PROBLEM STATEMENT

Online ticket reservation system is an efficient way to reserve tickets by standing in the long queue. Now, almost every reservation system has their own website for online reservation to provide better customer service. The manual filling of reservation form cannot be changed once the details had been entered.

The "Any Where Any Time Advance Reservation" Systems is the online ticket reserving systems where the passengers can reserve the tickets for their travel, cancel the reserved ticket and they can view the status of the ticket before travelling.

The passenger who is reserving the ticket in AWATAR unless they are the member of AWATAR. The passenger can reserve the ticket by ticket by giving the required detail in the form and submit it for processing. They can reserve for at the max for five members and a single ticket is provided for them.

The administrator the AWATAR can control the ticket reservation and the accounts of the passengers who are signing up in AWATAR. The administrator can reset the seats, fares of the tickets, and generates the TNR number for the ticket that are reserved.

The sole control of the system is handled by administrator. The printer prints the tickets that are issued by the passenger. The passenger can sign-up for only one time and he can sign in for any number of times for reserving, cancelling and viewing the tickets.

The AWATAR system provides flexibility for the persons based on the age, the passengers are fare based on the age and the place of travelling. This makes ease of use in using AWATAR system. The system track for the database any number of times for reserving, cancelling and status viewing.

The passenger can see the status of the classes that are available in the train which which he had is going to travel by noticing the number of seats available details from the display board.

The cancellation of the tickets is also very easy so that the passenger can cancel the tickets that he has booked.

## Experiment - 2

AIM → To write specification requirement sheet for any project

### THEORY

#### SRS (Software Requirement Specification)

as name suggests, is complete specification and description of any requirements of the software that needs to be fulfilled for successful development of software systems. These requirements can be functional as well as non-functional depending upon type of requirement. The interaction between different customers and contractor is done because it's necessary to fully understand needs of customers.

Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that is needed to be done to increase quality of product and to satisfy customer's demand.

#### PROBLEM REQUIREMENTS :

##### 1. Basic Requirements

- 1.a. Source place
- 1.b. Destination place
- 1.c. Date of journey

Date \_\_\_\_\_

Expt. No. \_\_\_\_\_

Page No. \_\_\_\_\_

## 2. Functional Requirements

- 2-a. Details of passengers
- 2-b. Details of the coach
- 2-c. Pay the amount using Visa card or debit card
- 2-d. Give out the balance.
- 2-e. Show the detail of ticket.

## 3. Non-Functional Requirements

- 3-a. Trading system failure
- 3-b. Unavailability of date
- 3-c. Coach unavailability
- 3-d. Insufficient amount for making payment.
- 3-e. Unavailability of bus.

Teacher's Signature \_\_\_\_\_

## EXPERIMENT-3

### Aim

To draw a use case diagram for any project.

### Theory

To draw use case diagram of finger f

### Use Case diagram

According to the UML specification use case diagram is a diagram that shows relationship between actors and use cases within a system. It is often used to communicate the development scope of a project.

#### 1. Use Cases :

A use case describe a sequence of action that provide measurable value to an actor. It is known drawn as a horizontal ellipse on UML use diagram.

#### 2. Actors :

An actor is a person that plays a role in one or more interaction with your system.

#### 3. System boundary Boxes -

The rectangle around the use cases is called the system boundary box as the name suggests it indicate the scope of your system. The use cases inside your system represent the functionality effect you intend to implement.

**Use case Diagram:**

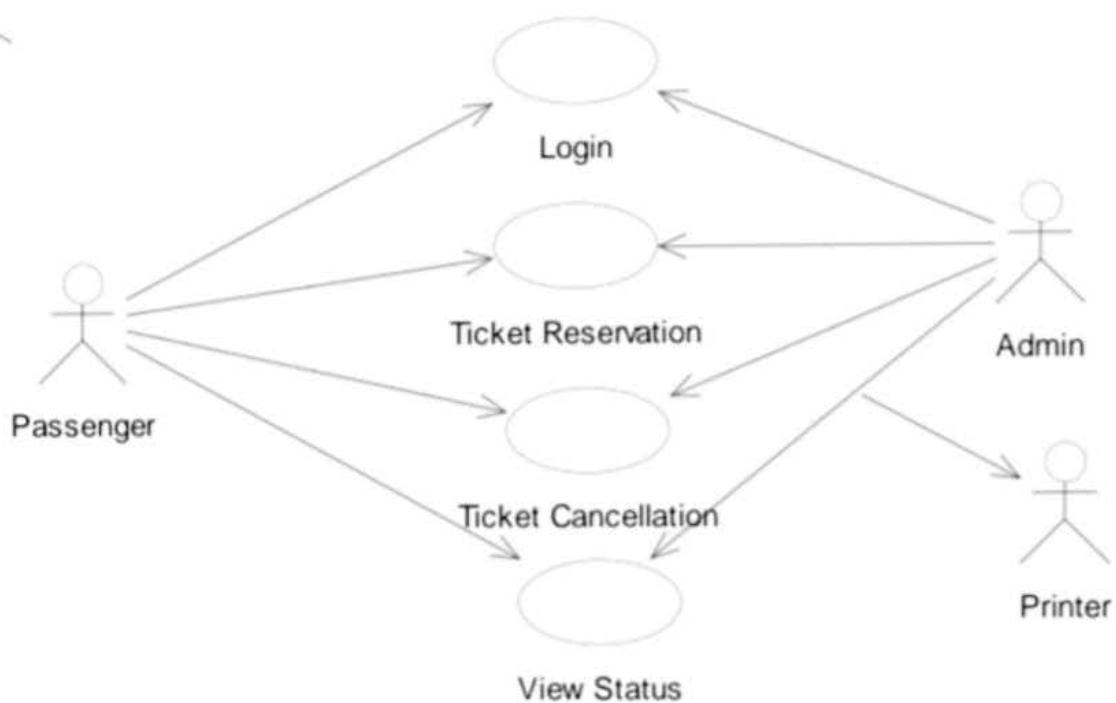


Figure 1. Use Case diagram for Online Ticket Reservation System

Date \_\_\_\_\_

Expt. No. \_\_\_\_\_

Page No. \_\_\_\_\_

4. Relationships :

Associations are depicted as linear connecting two - modeling elements with optional open-headed arrowhead on one end indicating the directions of initial invocation of relationship.

Teacher's Signature \_\_\_\_\_

## Experiment - 4

### Aim

To draw ER-model for any project.

### Theory

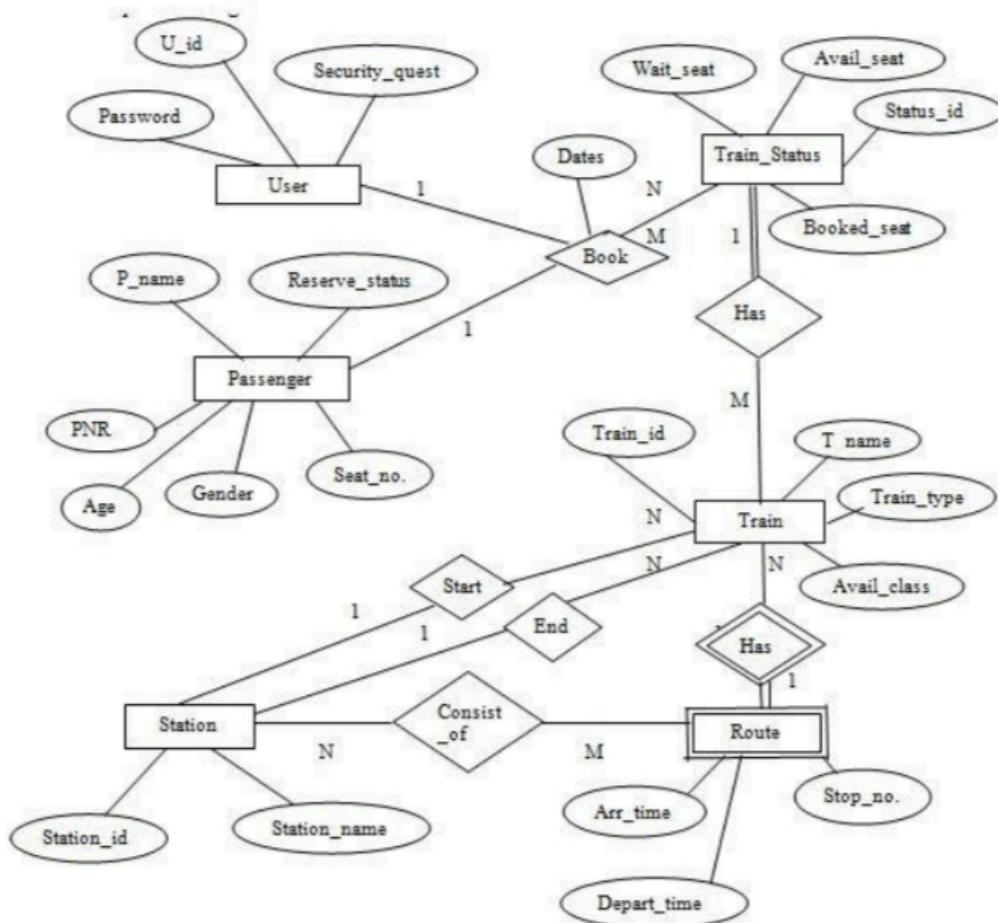
#### ER-model for

#### Entity Relationship diagram

Entity Relationship diagram are a major data modeling tool and will help to organize the data into entities & relationships

#### Entity Relation Component

1. Entity - An entity may be any object, class, person. In the ER diagram an entity can be represented as rectangle.
2. Attribute - The attribute is used to describe the property of an entity. Eclipse is used to represent it.  
There are 4 types of attribute :-
  1. Key attribute : represent main characteristic of entity.
  2. Composite attribute : attributes composed of many attributes.
  3. Multi valued attribute : attribute can have more than one value.
  4. Derived attribute : attribute derived from other attribute.
3. Relationship :- It is used to associate the relation b/w entities.
  1. One to One : Only one instance associates with the relationship.
  2. Many to Many : More than one instance on the left and more than one instance of an entity on right.



Date \_\_\_\_\_

Page No. \_\_\_\_\_

Expt. No. \_\_\_\_\_

3. Many to one: more than one instance of entity on left and only one instance on right.

4. One to many: One instance on left and more than one instance of an entity on right.

Teacher's Signature \_\_\_\_\_

## Experiment - 6

Aim - To draw activity diagram of any project

### Theory

An activity diagram is a behavioural diagram, it depicts the behaviour of a system. An activity diagram shows business & software process as a progression of actions.

### Activity Diagram Components:

1. Initial State: The starting state before an activity takes place using the initial state.
2. Activity State: An activity represents execution of an action on objects, we represent an activity using the initial state.
3. Action flow: They are used to make a decision before deciding the flow of control we use decision node.
4. Decision node: They are used to make a decision before deciding the flow of control we use decision node.
5. Fork: Fork nodes are used to support activities.
6. Join: Join nodes are used to support concurrent activities encouraging into one.

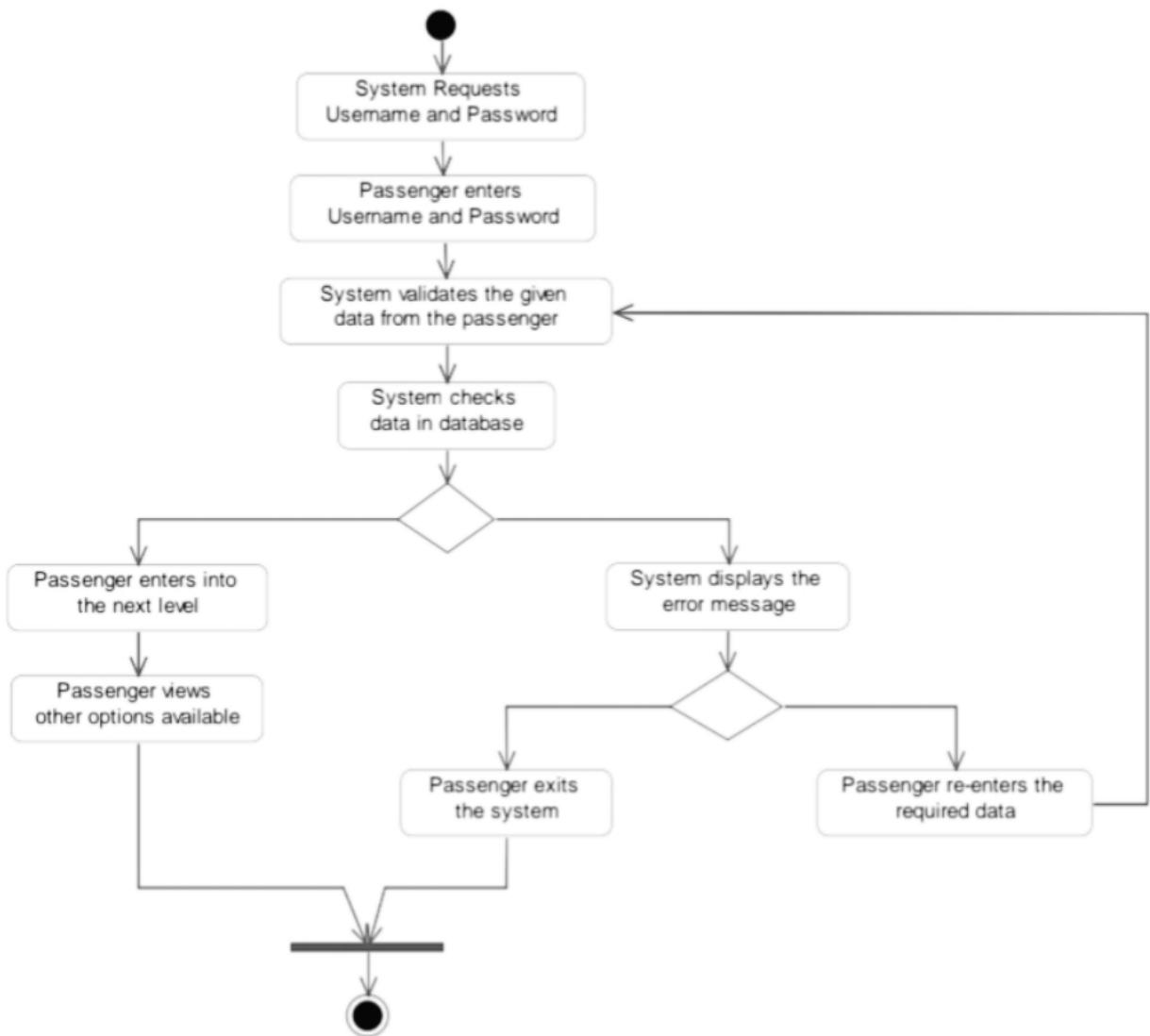


Figure 4. Activity Diagram for Login

Date \_\_\_\_\_

Expt. No. \_\_\_\_\_

Page No. \_\_\_\_\_

7. Merge : We can merge two or more activities into one if the control proceeds onto the non activity irrespective of path chosen.
8. Final state : The state at which particular activity stops. We use a filled circle within a circle notation to represent the final state.

Teacher's Signature \_\_\_\_\_



Figure 7. Activity diagram for Reservation

## Experiment-7

AIM: To draw sequence diagram for any project.

Sequence Diagram:-

It shows potential interaction between objects in the system being defined.

Normally these are specified as a part of use case to show how use case is implemented in the system.

Sequence diagram Components:

1. Lifeline → A lifeline represents an individual participant in the interaction.
2. Activation → A thin rectangle box is used to represent the time needed for object to complete task the longer the box the longer task it will take.
3. Guards → They are used to when we need to restrict the flow of messages on the present of a condition being met. They are important role in letting software developers know constraint attached to it.
4. Messages → Communication between objects is depicted using message.  
Messages can be classified as:

(i) Synchronous message: It requires a response before the interaction can continue. It is usually drawn using solid arrowhead pointing from one object to another.

(ii) Asynchronous message: It does not wait for the reply from the receiver.

### Sequence Diagram:

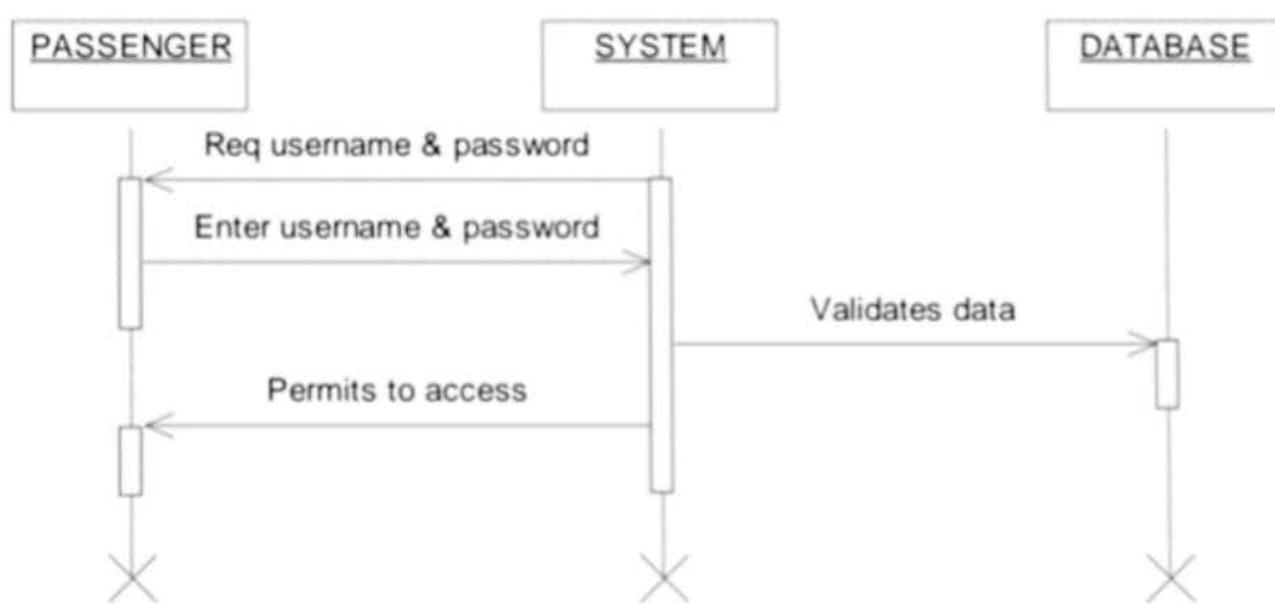


Figure 10. Sequence Diagram for Login

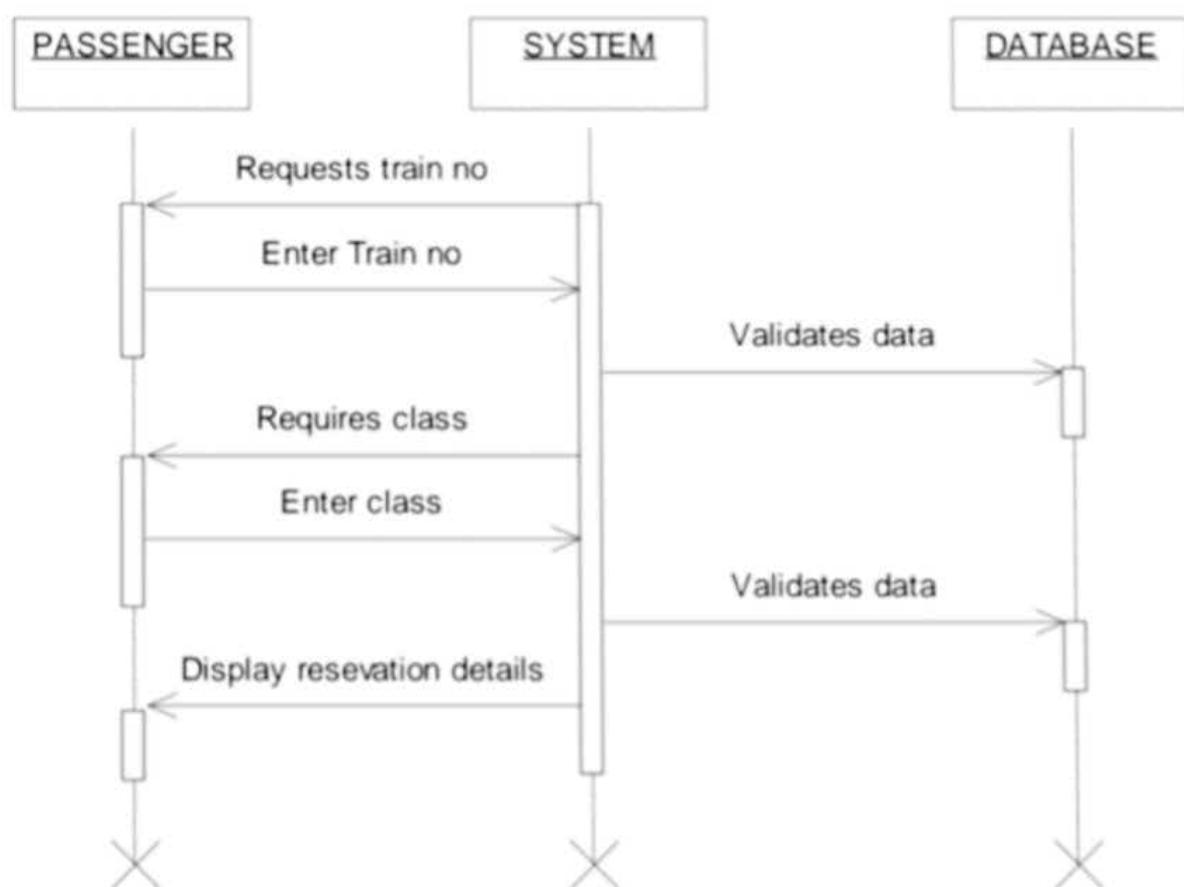


Figure 11. Sequence Diagram for Reservation

3. Create Message → It is used to instantiate a new object. It is represented with a dotted arrow.
4. Delete Message → It is used to delete an object.
5. Self Message → When object send message to itself.
6. Reply Message → They are used to show the message being sent from receiver to sender.

## EXPERIMENT-8

AIM → To draw class diagram of any project

### Class Diagram

Class diagram is a type of static structure diagram that describe the structure of a system by showing their attributes, operations & relationship among objects.

### Components

#### CLASS NOTATIONS !

1. Class Name → The name of class appears in first position.
  2. Class Attribute → Attributes are shown in second partition. The attribute type is shown after colon.
  3. Class Operation
- \* Operation are shown in third position  
 \* Operation map class methods in code.

#### CLASS RELATIONSHIP

A class may be involved in one or many relationships with other classes.

#### Types of Relationship !

1. Inheritance → Representation of "is-a" relationship. In this sub class 1 & sub class 2 are specialisation of super class.

Teacher's Signature \_\_\_\_\_

**Class Diagram:**

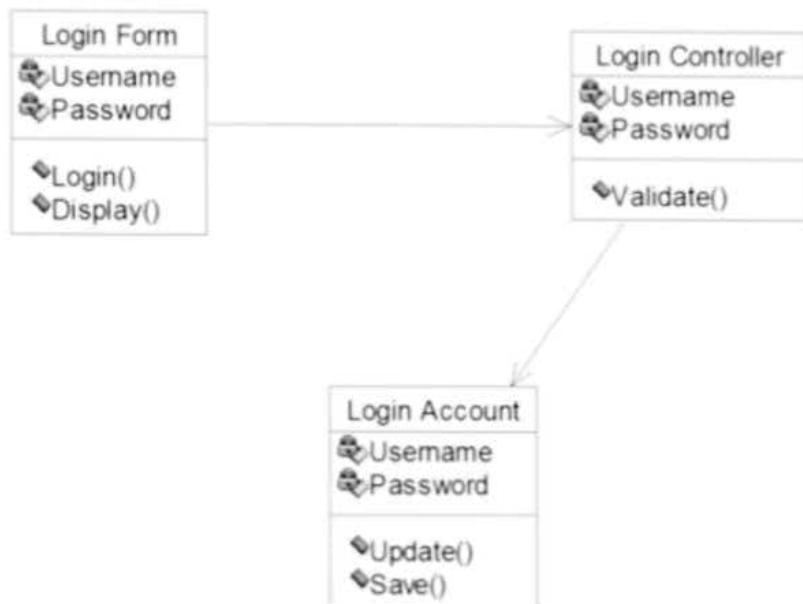


Figure 2.Class Diagram for Login

2. Simple association → There is an association between class 1 & class 2. A solid line connects these two classes.
3. Aggregation → It represents a "part of relationship" like class 2 is part of class 1. A solid line with an unfilled diamond at the association end connected to class of composite.
4. Composition → A special type of aggregation where parts are destroyed when the whole is destroyed like object class 2 lives & die with class 1.
5. Dependency → Exist between two classes if the changes to the definition of one cause changes to the other. A dashed line with open arrow is the way to represent it.

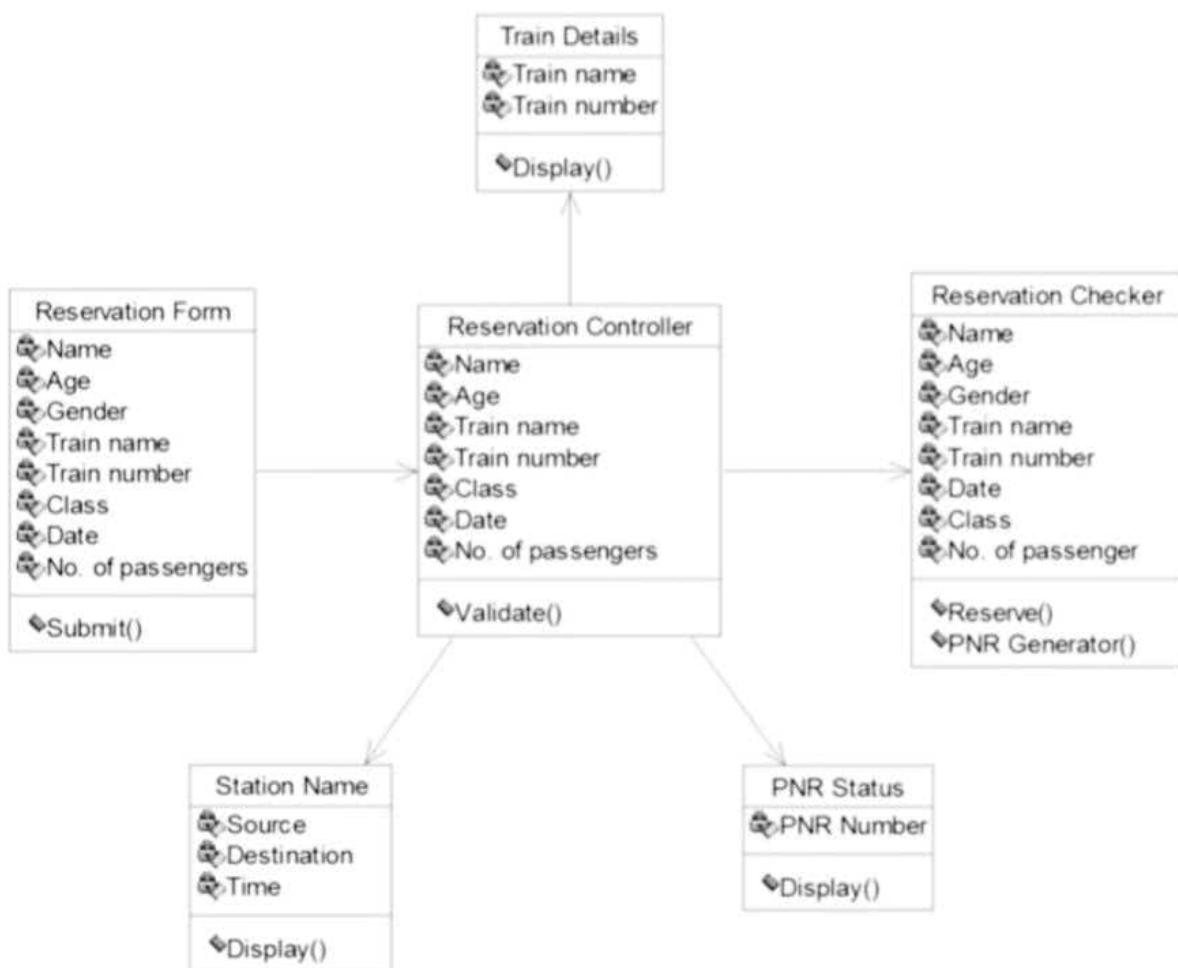


Figure 3. Class Diagram for Reservation

## EXPERIMENT - 9

AIM → To draw object diagram for any project.

## THEORY

Object diagrams are derived from class diagram so it dependent upon class diagram. Object diagrams are dependent represent instance of the class diagram. The basic concepts are similar for both diagram. It also represents the static view of a system but static view is a snapshot of system at a particular moment.

Components:

1. Object Names: Every object is symbolized like a rectangle that offers the name from the object of its class underlined as well & divided with colon.
2. Object Attributes: Similar to classes you are able to list object attribute inside separate component / compartment. However unlike classes, object attributes should have values assigned for them.
3. Links: Links tend to be instance associated with association. You can draw a link while using links utilized is class diagram.

## EXPERIMENT-10

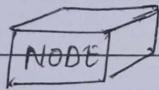
AIM → To draw deployment diagram of any project.

### Theory:-

**Deployment Diagram** :- It shows how a system will be physically deployed in the hardware environment. Its purpose is to show where the different components of the system will physically run & how they will communicate with each other. Since the diagram models the physical runtime, a system provider will make considerable use of this diagram.

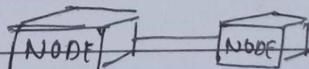
### Deployment Diagram Notations :-

#### 1. Nodes



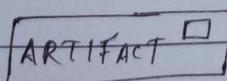
A node, represented as a cube is a physical entity that executes one or more components.

#### 2. Communication Association



This is represented by a solid line between two nodes. It shows both communication between nodes.

#### 3. Artifacts



are concrete elements that are caused by development process.

**Deployment Diagram:**

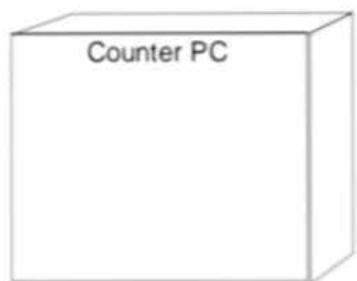
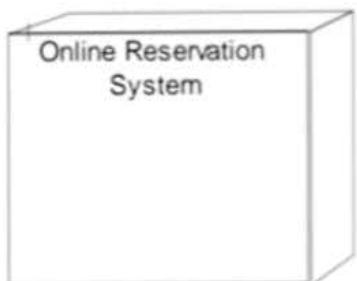
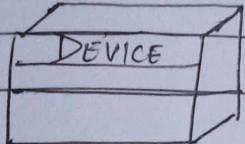


Figure 19. Deployment diagram for Online Ticket Reservation System

4. Device - A device is a node that is used to represent a physical computational resource in a system.



5. Deployment Specification : It is a configuration file such as a text file. It describes how an artifact is deployed on a node.

Deployment Specification