EX NO:6	E-TICKETING
DATE:	

AIM:

To draw the diagrams[use case, activity, sequence, collaboration, class, state chart, component, deployment, package] for the E-ticketing system.

SOFTWARE REQUIREMENTS SPECIFICATION

SL.NO	SOFTWARE REQUIREMENTS SPECIFICATION
1.0	Hardware Requirements
1.1	Software Requirements
1.2	Problem Analysis and Project Plan
1.3	Project description
1.4	Reference

1.0 HARDWARE REQUIREMENTS:

Intel Pentium Processor I3/I5

1.1 SOFTWARE REQUIREMENTS:

Rational rose / Argo UML

1.2 PROBLEM ANALYSIS AND PROJECT PLANNING

In the E-Ticketing system the main process is a applicant have to login the database then the database verifies that particular username and password then the user must fill the details about their personal details then selecting the flight and the database books the ticket then send it to the applicant then searching the flight or else cancelling the process

1.3 PROJECT DESCRIPTION:

This software is designed for supporting the computerized e-ticketing. This is widely used by the passenger for reserving the tickets for their travel. This E-ticketing is organized by the central system. The information is provided from the railway reservation system.

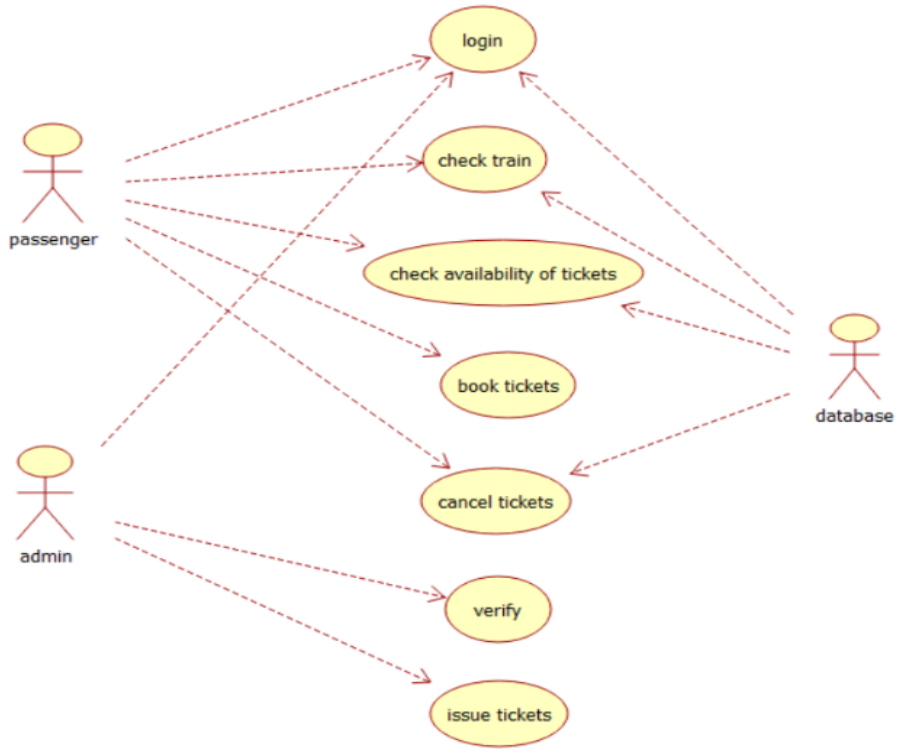
1.4 REFERENCES:

IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

This diagram will contain the actors, use cases which are given below

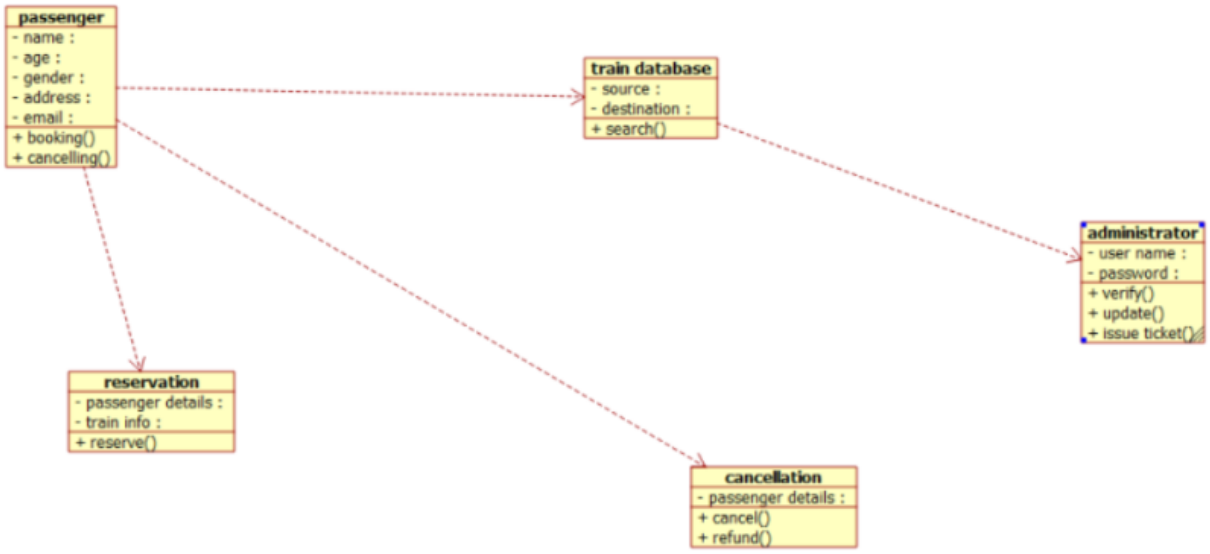
Actors: Passenger, Ticketing Agent, Payment Gateway, System Administrator.
Use case: User Registration, Login, Search Tickets, Select Travel Details, Book Ticket, Make Payment, Generate E-Ticket, Cancel Ticket, View Booking History



CLASS DIAGRAM:

This diagram consists of the following classes, attributes and their operations.

CLASSES	ATTRIBUTES	OPERATIONS
User	User ID,Name,Email	Login(),Logout(),View profile()
Passenger	Passenger ID,Age,Seat preference	Search ticket(),Book ticket(),View booking status()
Ticket	Ticket ID,Journey date,Seat number	Generate ticket(),Cancel ticket(),View ticket details()
Payment	Payment ID,Amount,Payment status	Make payment(),Refund amount(),View payment status()
Admin	Admin ID,Role,Contact no	Manage schedule(),Verify ticket(),Generate report()



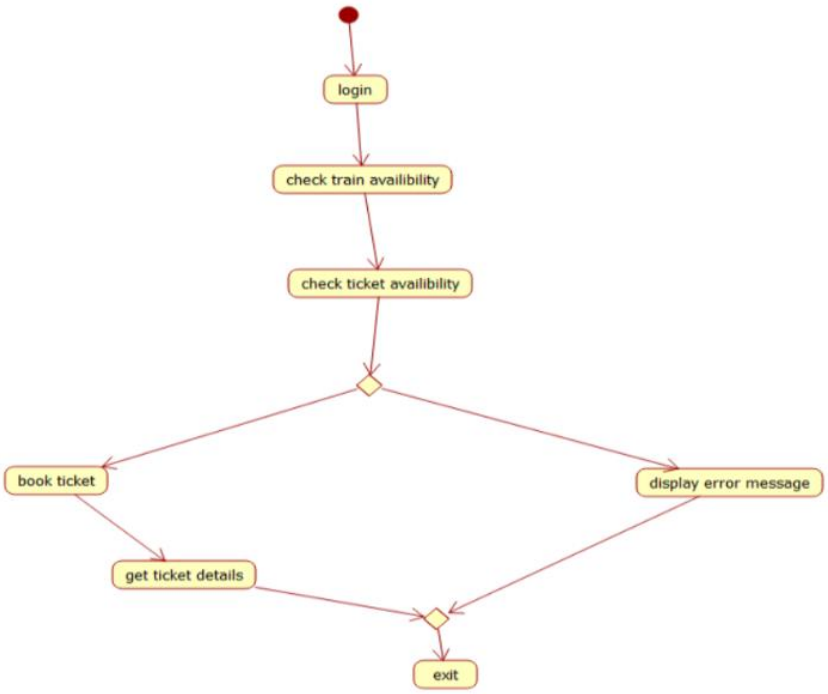
ACTIVITY DIAGRAM:

below:

This diagram will have the activities as Start point, End point, Decision boxes as given

Activities: enter the train number, enter the number of seats, acceptance of ticket, accept seat.

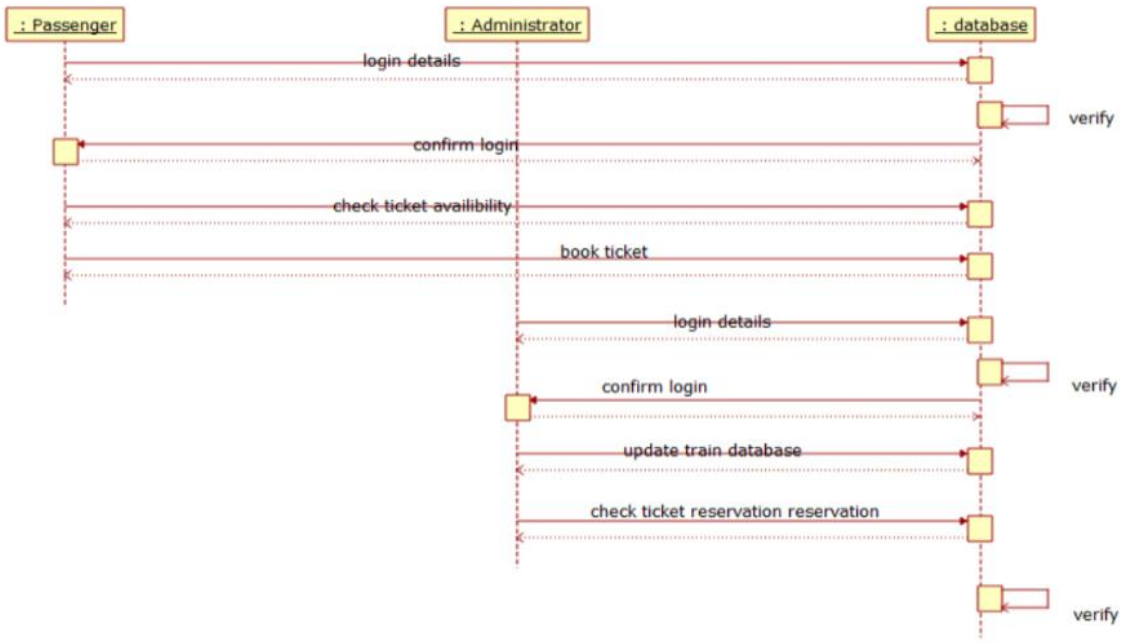
Decision box: Check availability of seats whether it is present or not.



SEQUENCE DIAGRAM:

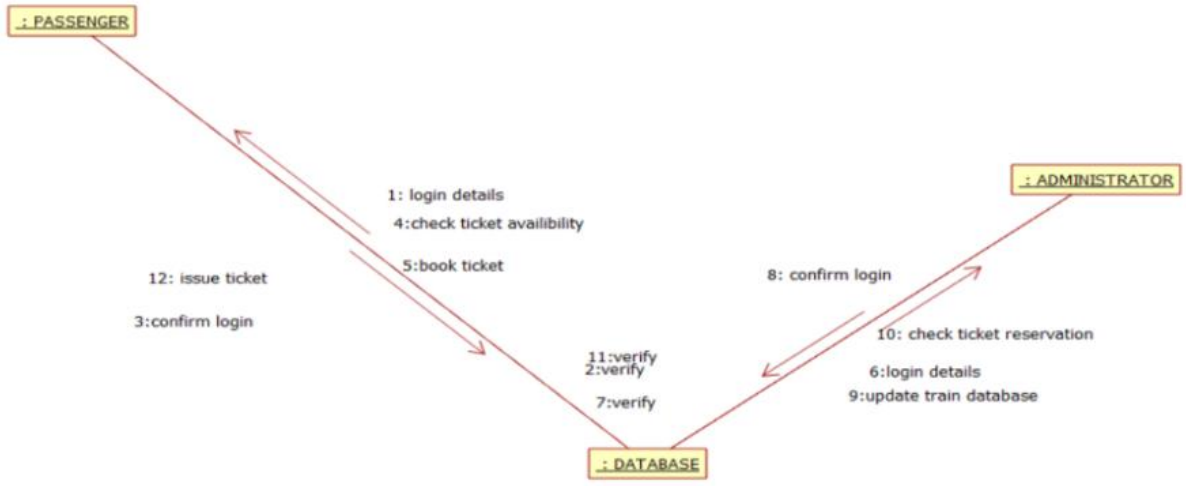
This diagram consists of the objects, messages and return messages.

Object: Passenger, Railway reservation system, Central computer.



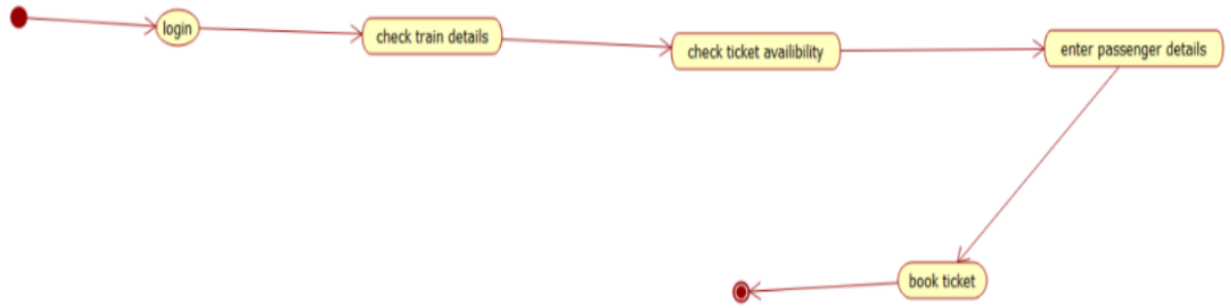
COLLABORATION DIAGRAM:

This diagram contains the objects and actors. This will be obtained by the completion of the sequence diagram and pressing the F5 key.



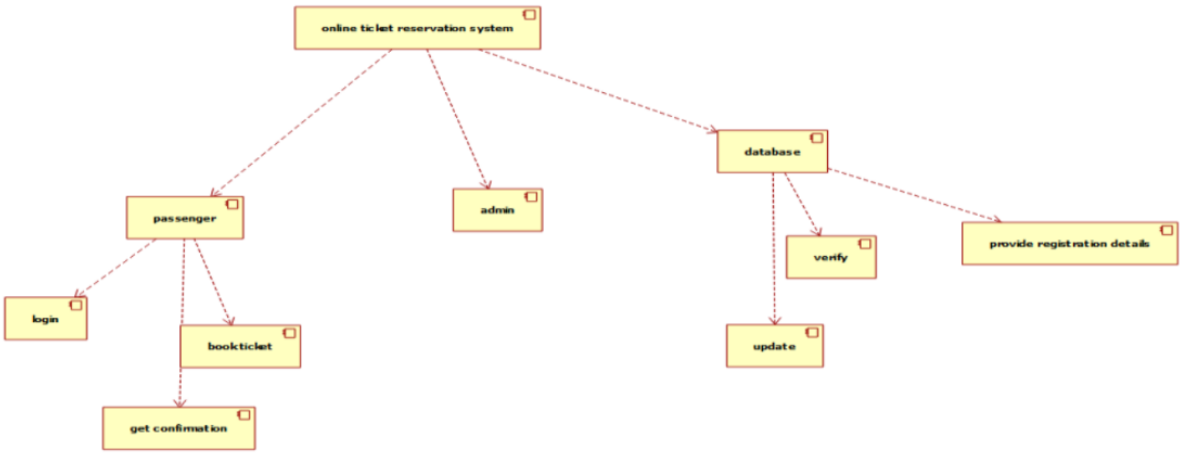
STATE CHART DIAGRAM:

It is a technique to describe the behavior of the system. It describes all the possible states that a particular object gets into the object oriented technique. State diagram are drawn for a single class to show to the lifetime behaviour of a single objects



COMPONENT DIAGRAM:

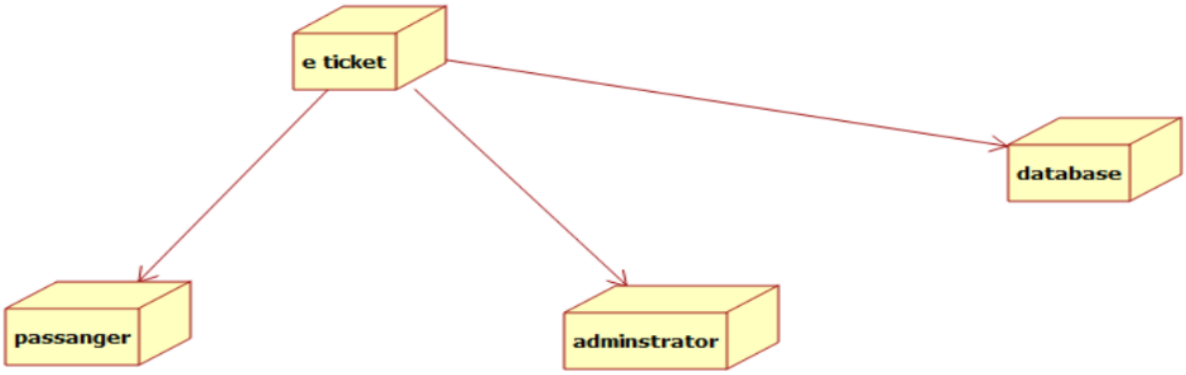
The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication association.



DEPLOYMENT DIAGRAM:

A deployment diagram in the unified modeling language serves to model the physical

deployment of artifacts on deployment targets. Deployment diagrams show "the allocation of artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimentional box. Dependencies are represented by communication association.

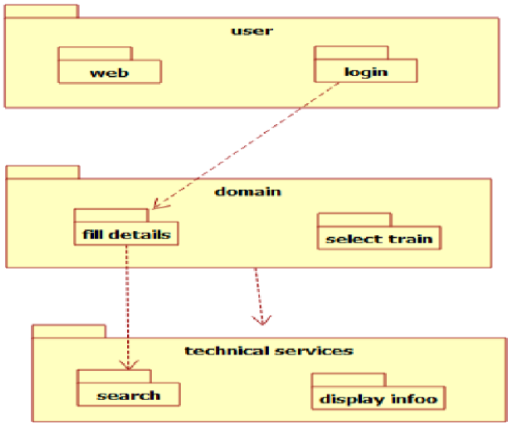


PACKAGE DIAGRAM:

A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs).

There are three types of layer. They are

- o User interface layer
- o Domain layer
- o Technical services layer



PROGRAM CODING:

PASSENGER:

```
Public class passenger
{
        Public integer passenger passenger name;
        Public integer passenger passenger age;
        Public integer train no;
        Public void passenger()
        {
        }

Public void new operation()
}
}
```

CENTRAL MANAGEMENT SYSTEM:

```
Public class central management
{
        Public integer train name;
        Public integer passenger name;
        Public void reservation()
        {
        }
        Public void cancellation()
        {
        }
        Public void status()
        {
        }
        Public void login()
```

```
        {
        }
        Private void management()
        {
        }
}
```

RAILWAY RESERVATION SYSTEM:

```
Public class railway reservation system
{
Public integer trainno; Public
integer train name;
        Public integer passenger name;
        Public void status()
        {
        }
        Public void reservation()
        {
        }
        Public void cancellation()
        {
        }
        Public void railway reservation system()
        {
        }
}
```

RESULT:

Thus the diagrams[use case, activity, sequence, collaboration, class, statechart, component, deployment, package] for the E-ticketing system has been designed, executed and output is verified.