

Kameshwaran. K (192421120)

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.

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.

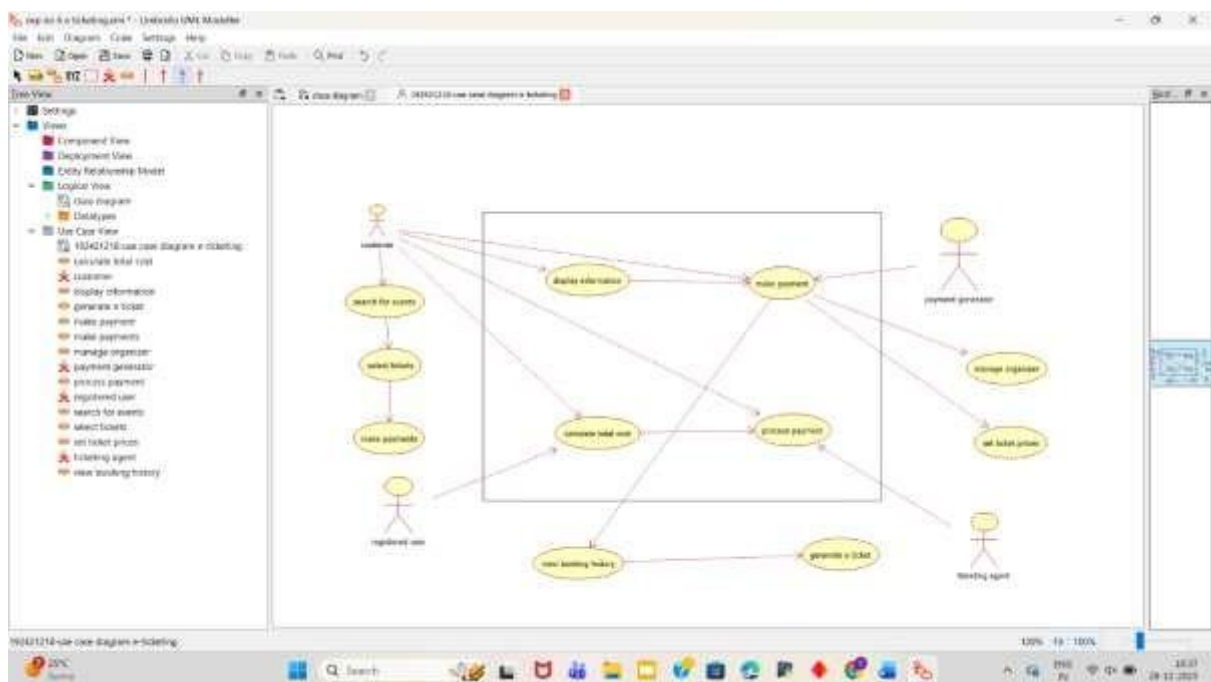
IEEE Software Requirement Specification format.

USE CASE DIAGRAM:

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

Actors: Passenger, Railway reservation system..

Use case: Status, reservation, cancellation, enter the train number, enter the number of seats, availability of seats, acceptance of ticket.



CLASS DIAGRAM:

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

CLASSES ATTRIBUTES OPERATIONS

Central computer Train name,

Passenger name

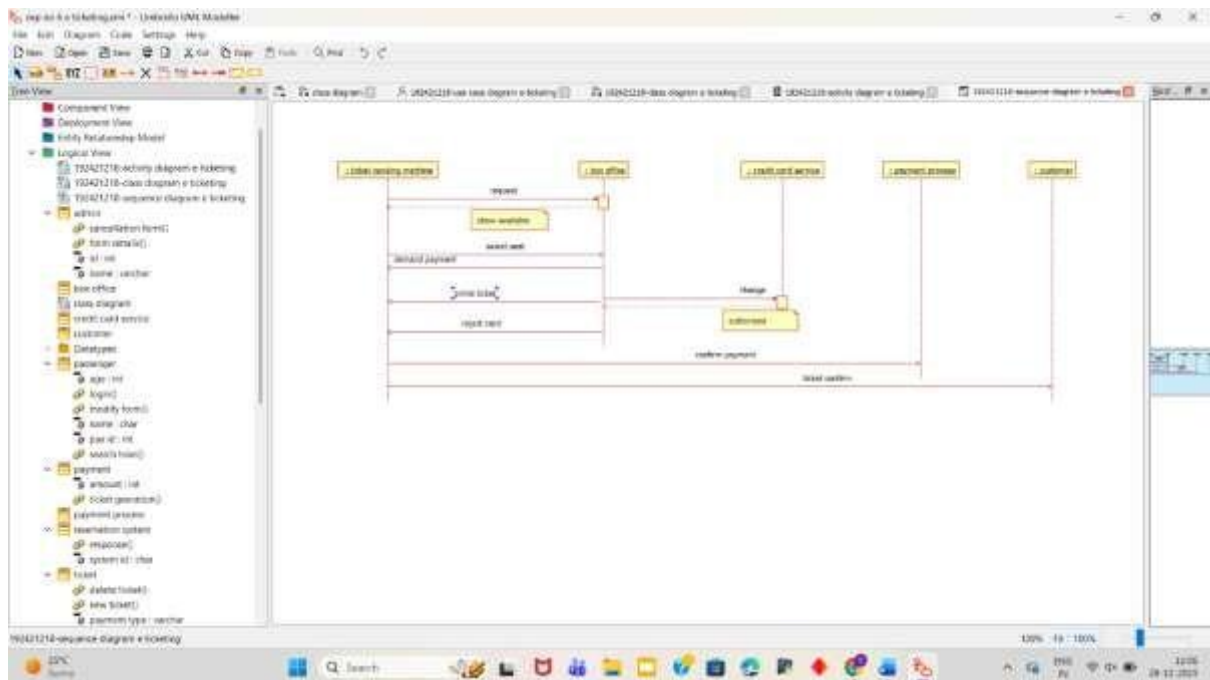
Reservation(), login()

Passenger Passenger age Login()

Railway reservation system Train number Cancellation()

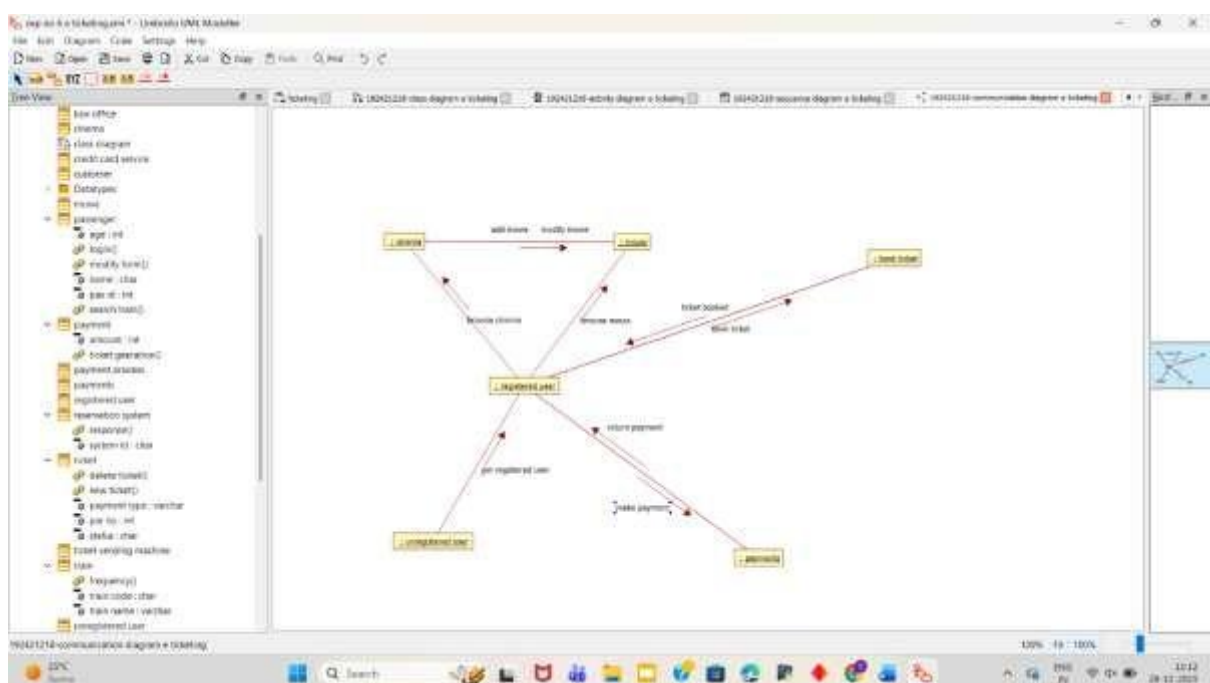
This diagram consists of the objects, messages and return messages.

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

The screenshot shows the UML Modeler application interface. On the left, a project tree lists various UML elements: actor, usecase, class, sequence diagram, etc. The main workspace displays a sequence diagram with the following elements:

- Start Node**: A red dot representing the start of the process.
- sign to register**: A yellow rectangular node.
- select name**: A label on the arrow connecting 'sign to register' to 'booked name'.
- booked name**: A yellow rectangular node.
- searched for booking**: A yellow rectangular node.
- cancel**: A label on the arrow connecting 'booked name' to 'searched for booking'.
- release trial**: A yellow rectangular node.
- cancel booking**: A yellow rectangular node.
- End Node**: A red dot representing the end of the process.

The diagram shows a flow from 'sign to register' to 'booked name' (labeled 'select name'), then to 'searched for booking' (labeled 'cancel'). From 'searched for booking', there are two paths: one to 'release trial' (labeled 'no') and another to 'cancel booking' (labeled 'yes'). Both 'release trial' and 'cancel booking' lead to the final end node.

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

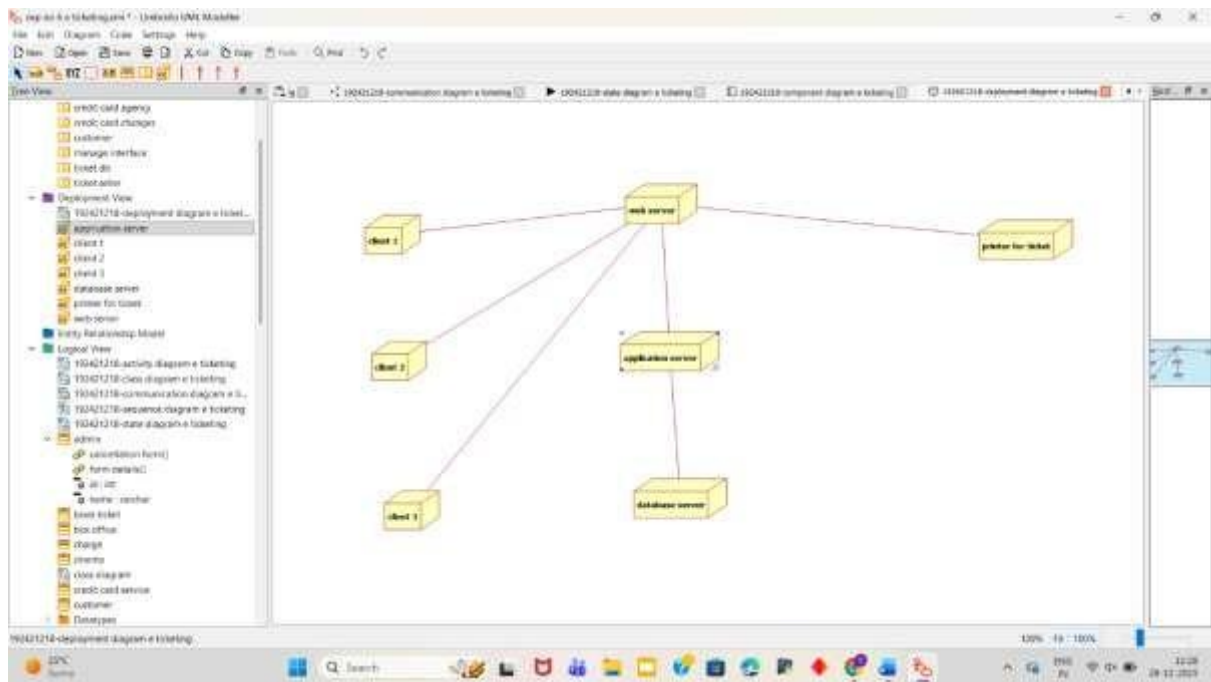
The screenshot displays the UML Modeler interface with a sequence diagram titled "190421218 sequence diagram v ticketing". The diagram illustrates the following sequence of interactions:

- customer** (actor) sends a message to **check** (object).
- check** sends a message to **group info** (object).
- check** sends a message to **individual info** (object).
- group info** sends a message to **ticket seller** (object).
- individual info** sends a message to **ticket seller** (object).
- ticket seller** sends a message to **credit card changes** (object).
- credit card changes** sends a message to **credit card expiry** (object).
- ticket seller** sends a message to **ticket ok** (object).
- manage interface** (object) sends a message to **ticket ok** (object).

The left sidebar shows the project structure, including a package named "190421218" containing various UML diagrams. The bottom status bar indicates the diagram is a "190421218-component diagram v ticketing" and shows system information like CPU usage and date.

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- dimensional box. Dependencies are represented by communication association.



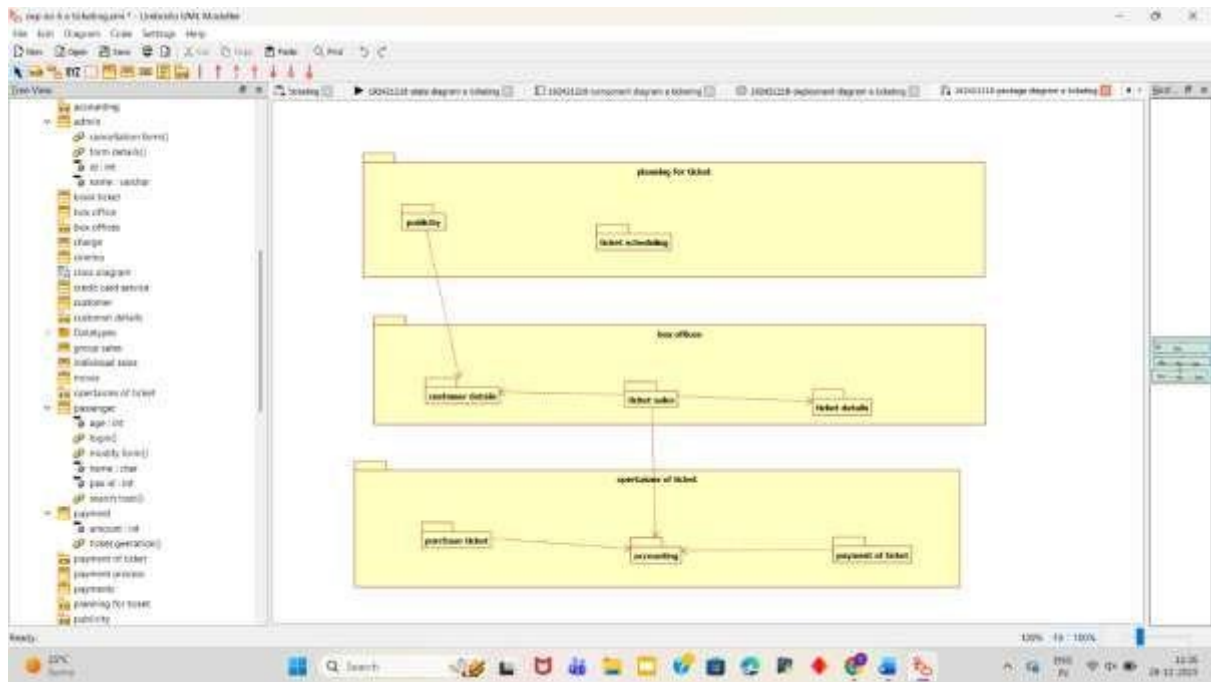
PACKAGE DIAGRAM:

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



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.