

VENDING MACHINE

PROJECT REPORT

18CSC202J/ 18AIC203J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY

(2018 Regulation)

II Year/ III Semester

Academic Year: 2022 -2023

By

VYSHNAV VEERAVALLI (RA2111026010235)

MOHITH JAYAVARAM (RA2111026010241)

Under the guidance of

Dr. M. Uma

Assistant Professor

Department of Computational Intelligence



FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur

NOVEMBER 2022

BONAFIDE

This is to certify that **18CSC202J - OBJECT ORIENTED DESIGN AND PROGRAMMING LABORATORY project report** titled “**RAILWAY RESERVATION SYSTEM**” is the bonafide work of **VYSHNAV VEERAVALLI (RA2111026010235) & MOHITH JAYAVARAM (RA2111026010241)**

who undertook the task of completing the project within the allotted time.

Signature of the Guide

Dr. M. Uma

Assistant Professor

Department of CINTEL,

SRM Institute of Science and Technology

Signature of the II Year Academic Advisor

Professor and Head

Department of CINTEL

SRM Institute of Science and Technology

About the course:-

18CSC202J/ 8AIC203J - Object Oriented Design and Programming are 4 credit courses with **L T P C as 3-0-2-4** (Tutorial modified as Practical from 2018 Curriculum onwards)

Objectives:

The student should be made to:

- Learn the basics of OOP concepts in C++
- Learn the basics of OOP analysis and design skills.
- Be exposed to the UML design diagrams.
- Be familiar with the various testing techniques

Course Learning Rationale (CLR): The purpose of learning this course is to:

- 1.Utilize class and build domain model for real-time programs
- 2.Utilize method overloading and operator overloading for real-time application development programs
- 3.Utilize inline, friend and virtual functions and create application development programs
- 4.Utilize exceptional handling and collections for real-time object-oriented programming applications
- 5.Construct UML component diagram and deployment diagram for design of applications
- 6.Create programs using object-oriented approach and design methodologies for real-time application development

Course Learning Outcomes (CLO): At the end of this course, learners will be able to:

- 1.Identify the class and build domain model
- 2.Construct programs using method overloading and operator overloading
- 3.Create programs using inline, friend and virtual functions, construct programs using standard templates
- 4.Construct programs using exceptional handling and collections
- 5.Create UML component diagram and deployment diagram
- 6.Create programs using object-oriented approach and design methodologies

Table 1: Rubrics for Laboratory Exercises

(Internal Mark Split-up :- As per Curriculum)

CLAP-1	5=(2(E-lab Completion) + 2(Simple Exercises)(from CodeZinger, and any other coding platform) + 1(HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-2	7.5=(2.0(E-lab Completion)+ 2.0 (Simple Exercises)(from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	Elab test
CLAP-3	7.5=(2.0(E-lab Completion(80 Pgms)+ 2.0 (Simple Exercises)(from CodeZinger, and any other coding platform) + 3.5 (HackerRank/Code chef/LeetCode Weekend Challenge)	2 Mark - E-lab Completion 80 Program Completion from 10 Session (Each session min 8 program) 2 Mark - Code to UML conversion GCR Exercises 3.5 Mark - Hacker Rank Coding challenge completion
CLAP-4	5= 3 (Model Practical) + 2(Oral Viva)	<ul style="list-style-type: none"> • 3 Mark – Model Test • 2 Mark – Oral Viva
Total	25	

COURSE ASSESSMENT PLAN FOR OODP LAB

S.No	List of Experiments	Course Learning Outcomes (CLO)	Blooms Level	PI	No of Programs in each session
1.	Implementation of I/O Operations in C++	CLO-1	Understand	2.8.1	10
2.	Implementation of Classes and Objects in C++	CLO-1	Apply	2.6.1	10
3,	To develop a problem statement. 1. From the problem statement, Identify Use Cases and develop the Use Case model. 2. From the problem statement, Identify the conceptual classes and develop a domain model with a UML Class diagram.	CLO-1	Analysis	4.6.1	Mini Project Given
4.	Implementation of Constructor Overloading and Method Overloading in C++	CLO-2	Apply	2.6.1	10
5.	Implementation of Operator Overloading in C++	CLO-2	Apply	2.6.1	10
6.	Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams and Collaboration diagrams	CLO-2	Analysis	4.6.1	Mini Project Given
7.	Implementation of Inheritance concepts in C++	CLO-3	Apply	2.6.1	10
8.	Implementation of Virtual function & interface concepts in C++	CLO-3	Apply	2.6.1	10
9.	Using the identified scenarios in your project, draw relevant state charts and activity diagrams.	CLO-3	Analysis	4.6.1	Mini Project Given
10.	Implementation of Templates in C++	CLO-3	Apply	2.6.1	10
11.	Implementation of Exception of Handling in C++	CLO-4	Apply	2.6.1	10
12.	Identify the User Interface, Domain objects, and Technical Services. Draw the partial layered, logical architecture diagram with UML package diagram notation such as Component Diagram, Deployment Diagram.	CLO-5	Analysis	4.6.1	Mini Project Given
13.	Implementation of STL Containers in C++	CLO-6	Apply	2.6.1	10
14.	Implementation of STL associate containers and algorithms in C++	CLO-6	Apply	2.6.1	10

15.	Implementation of Streams and File Handling in C++	CLO-6	Apply	2.6.1	10
-----	--	-------	-------	-------	----

LIST OF EXPERIMENTS FOR UML DESIGN AND MODELLING:

To develop a mini-project by following the exercises listed below.

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant state charts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.

Suggested Software Tools for UML:

StarUML, Rational Suite, Argo UML (or) equivalent, Eclipse IDE and Junit

RAILWAY RESERVATION SYSTEM

Abstract :

The Railway Reservation System facilitates the passengers to enquire about the trains available on the basis of source and destination, Booking and Cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of case study is to design and develop a database maintaining the records of different trains, train status, and passengers.

This project contains Introduction to the Railways reservation system .It is the computerized system of reserving the seats of train seats in advanced. It is mainly used for long route. On-line reservation has made the process for the reservation of seats very much easier than ever before.

In our country India, there are number of counters for the reservation of the seats and one can easily make reservations and get tickets. Then this project contains entity relationship model diagram based on railway reservation system and introduction to relation model .There is also design of the database of the railway reservation system based on relation model. Example of some SQL queries to retrieves data from rail management database.

1. Introduction:

PURPOSE

The purpose of this source is to describe the railway reservation system, which provides the train timing details, reservation, billing and cancellation on various types of reservation namely. In addition, This Railway reservation service will not only enhance the reservation but will also help the commuters in getting support, refunds and other real time fixes.

Confirm Reservation for confirm Seat.

Reservation against Cancellation.

Waiting list Reservation.

Online Reservation.

TatkalReservation

SCOPE

Technology has transformed many aspects of life in the 21st century, including the way many of us make train

reservations. For example, to make ticketing more convenient for travelers, an online reservation system helps us in booking tickets from the comfort of our homes or offices. While this is convenient for most people, it has made things particularly easier for people residing in remote locations.

The various advantages of using the online reservation system are as follows:

Convenient – You can book or cancel your tickets sitting in the comfort of your home or office.

Saves Time and Effort - You can save the time needed to travel to the railway reservation office and waiting in the queue for your turn.

Towards a greener planet – Instead of printing your ticket you can also choose to travel with the soft copy of your booked ticket in your laptop or even on your mobiles
Freight Revenue enhancement.

Passenger Revenue enhancement.

Improved & optimized service

DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

Some of the few ACRONYMS, ABBREVIATIONS used in Ticket Reservation System of Indian Railways are as follows:

NTES – National Train Enquiry System

IVRS – Interactive Voice Response system

PRS–passengerreservationsystem

RAC-Reservation Against Cancellation

OVERVIEW

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, the functional requirements, data requirements and constraints and assumptions made while designing or development of a Railway Reservation System. It also gives the user's viewpoint of the product. Section 3 gives the detail of all type of specific requirements of this product. It also includes the external, internal requirements and gives detailed description of functional requirements. However,

Section 4 of this Document is of completely supporting information, same is the case with LastSection

(Appendices). In Other words, below are the details of all the sections respectively:

PRODUCT PERSPECTIVE:

Before making this a real time running online reservation system, old system suffered from many of the DRAWBACKS, such as:

The existing system is highly manual involving a lot of paper workand calculation and therefore may be erroneous. This has led to inconsistency and inaccuracy in the maintenance of data.

The data, which is stored on the paper only, may be lost, stolen or destroyed due to any natural calamity of fire or water.

Existing system is sluggish and consumes a lot of time, resource etc. causing inconvenience to customers and staff.

Due to manual working, it is difficult to add, delete, update, or viewthe data.

Since number of passengers has increased to an uncertain multiple, it is very difficult to maintain or retrieve detailed record of passengers.

A Railway has many offices around the world, an absence of link between them all causes to a lack of miscommunication and discoordination.

Hence, this Railway reservation system is proposed, with following benefits:

Computerization of reservation system will reduce a lot of daily paperwork and hence load on the staff of admin department.

Machine does all the calculations. Hence, chances of error are low

Reservation, Cancellation or updation lists of Ticket's can easily be maintained and retrieved and any required additions, deletion or updation can easily be performed.

This system provides User Name-Password validation, hence unauthorized access is prevented

PRODUCT FUNCTIONS:

Users with varying levels of familiarity with computers will mostly use this system. With this in mind, an important feature of this software is that it can be

relatively simple to use. The scope of this project encompasses:

Search: This function allows the user to search for train that is available between two travel cities namely “Departure City” and “Arrival City” as desired by the traveler. The System initially prompts the agent for these two column values, the date of the journey, preferred time slot and the number of passengers. It then displays a list of trains available with different classes.

Selection: The function allows a particular train to be selected from the displayed list. All the details of the train are as shown:

Train Number.

Date, Time and place of departure.

Train Duration

Fair per head

Number of stoppages – 0, 1, 2

Review: if the seats are available, the software prompts for the booking of train. The train information is shown, The total fare including taxes is shown and train details are reviewed before final payment

Travel Information: This system asks for details of all the passenger before the booking confirmation. Hence, lesser cases of seat issues.

Payment: It requires details of credit/ debit card of the person to make payment and reserve the required seat, Details such as:

Card Number

Card Type

CVV Number

Expiry Date

Name on card.

Cancellation: System also allow cancellation of existing reservation done, making seats unreserved for others to book and refunding the money back to the accounts of users cancelling the tickets.

RAILWAY RESERVATION SYSTEM

Description : The railway reservation system facilitates the passengers to enquiry about the trains available on the basis of source and destination, booking and cancellation of tickets, enquiry about the status of the booked ticket, etc. The aim of case study is to design and develop a data base maintaining records of different trains, train status and passengers. This project contains introduction to the railways reservation system. It is the computerized system of reserving the seats of train seats in advance. It is mainly used for a long route. Online reservation has made the process for the reservation of seats very much easier than ever before. By selecting "Ticket Reservation," the user must enter their name, quantity of tickets, and journey train. The system will then request ticket confirmations. The user can also get a comprehensive list of all available trains. After purchasing a ticket, you have the option to cancel your reservation. This entire project is programmed in C++.

```
graph LR
    Actor1((Actor1)) --- UC11([manage users])
    Actor1 --- UC12([manage booking])
    Actor1 --- UC13([manage customers])
    Actor1 --- UC14([login and logout])
    Actor1 --- UC15([update profile])
    Actor1 --- UC16([manage payment])
    Actor1 --- UC17([manage ticket])
    Actor1 --- UC18([manage train schedule])
    Actor2((Actor2)) --- UC18
    Actor2 --- UC17
    Actor3((Actor3)) --- UC21([book ticket])
    Actor3 --- UC22([vacant seat])
    Actor3 --- UC23([payment])
    Actor3 --- UC15
    Actor8((Actor8)) --- UC24([book ticket])
    Actor8 --- UC25([make payment])
    Actor8 --- UC26([change password])
    Actor8 --- UC27([search trains])
    UC12 --- UC13
    UC13 --- UC14
    UC14 --- UC15
    UC15 --- UC16
    UC16 --- UC17
    UC17 --- UC18
    UC21 --- UC22
    UC22 --- UC23
    UC23 --- UC24
    UC24 --- UC25
    UC25 --- UC26
    UC26 --- UC27
```


EXPLANATION :

This Use Case Diagram is a graphic depiction of the interactions among the elements of Railway Reservation System. It represents the methodology used in system analysis to identify, clarify, and organize system requirements of Railway Reservation System. The main actors of Railway Reservation System in this Use Case Diagram are: Super Admin, System User, Ticket Agent,

Customers, who perform the different type of use cases such as

Manage Train, Manage Ticket, Manage Booking, Manage Customer, Manage Payment, Manage Train Route, Manage Train Schedule, Manage Users and Full Railway Reservation System Operations. Major elements of the UML use case diagram of Railway Reservation System are shown on the picture below.

The relationships between and among the actors and the use cases of Railway Reservation System:

Super Admin Entity: Use cases of Super Admin are Manage Train,

Manage Ticket, Manage Booking, Manage Customer, Manage

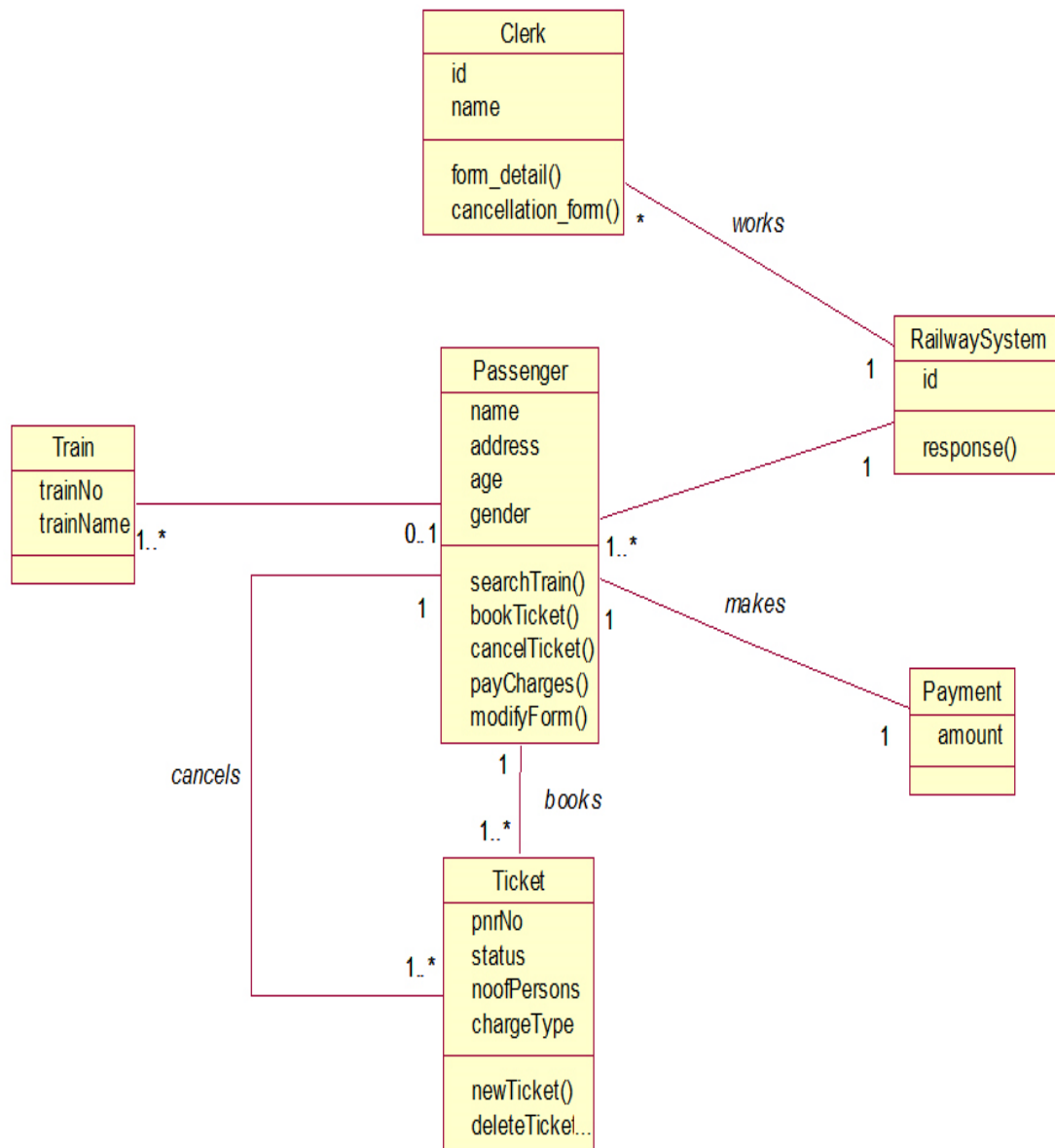
Payment, Manage Train Route, Manage Train Schedule, Manage Users and Full Railway Reservation System Operations

System User Entity: Use cases of System User are Manage Train, Manage Ticket, Manage Booking, Manage Customer, Manage Payment, Manage Train Route, Manage Train Schedule

Ticket Agent Entity: Use cases of Ticket Agent are Book Tickets, Search Vacant Seats, Collect Payment

Customers Entity: Use cases of Customers are Search Trains, Book Tickets, Make Payments

CLASS DIAGRAM



EXPLANATION :

- . Train Class: Manage all the operations of Train
- Ticket Class: Manage all the operations of Ticket
- Booking Class: Manage all the operations of Booking
- . Customer Class: Manage all the operations of Customer
- Payment Class: Manage all the operations of Payment •
- Train Route Class: Manage all the operations of Train Route

Classes and their attributes of Railway Reservation System Class Diagram:

. Train Attributes: train_id, train_name, train_number, train_seat_number, train_ticket, train_type, train_description

• Ticket Attributes: ticket_id, ticket_customer_id, ticket_type, ticket_date, ticket_description

Booking

Attributes: booking_id, booking_title, booking_type, booking_ticket, booking_date, booking_description

• Customer Attributes: customer_id, customer_name, customer_mobile, customer_email, customer_username, customer_password, customer address

Payment Attributes: payment_id, payment_customer_id, payment_date, payment_amount, payment_description

Train Route Attributes: train_route_id,
train_route_name, train_route_type,
train_route_description

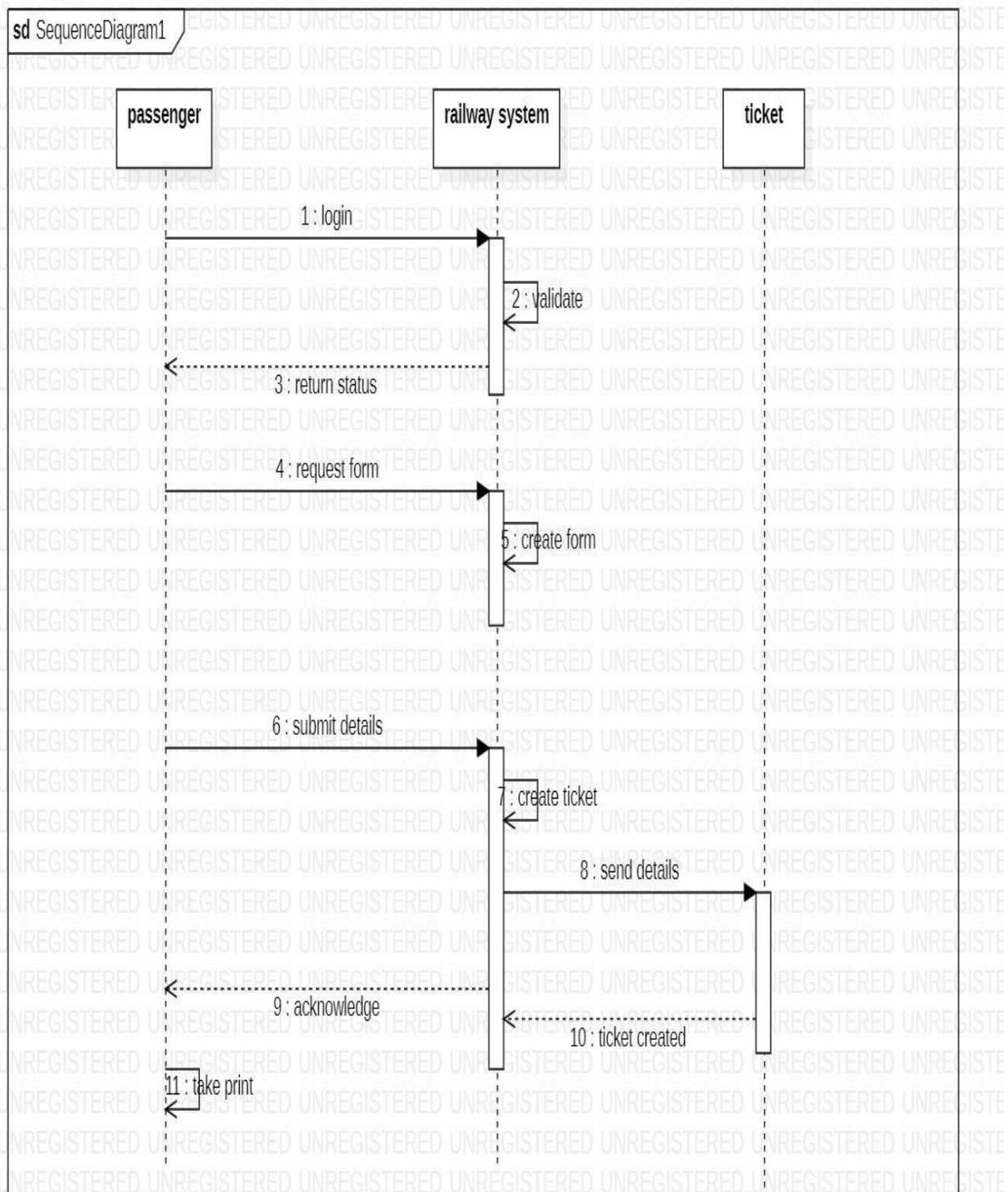
Train Methods: addTrain(), editTrain(), delete Train(),
update Train(), save Train(), searchTrain() • Ticket
Methods: add Ticket(), editTicket(), delete Ticket(),
update Ticket(), save Ticket(), search Ticket()

Booking Methods: addBooking(), editBooking(),
deleteBooking(), updateBooking(), saveBooking(),
searchBooking()

Customer Methods: addCustomer(),
editCustomer(), deleteCustomer(),
updateCustomer(), saveCustomer(),
searchCustomer()

Payment Methods: addPayment(), editPayment(),
deletePayment(), updatePayment(), savePayment(),
searchPayment() Train Route Methods: addTrain
Route(), editTrain Route(), delete Train Route(),
update Train Route(), save Train Route(), search Train
Route()

SEQUENCE DIAGRAM



EXPLANATION:

This is the Login Sequence Diagram of Railway Reservation System, where admin will be able to login in their account using their credentials. After login user can manage all the operations on Train Route, Customer, Ticket, Booking. Train Schedule. All the pages such as Ticket, Booking, Train Schedule are secure and user can access these page after login. The diagram below helps demonstrate how the login page works in a Railway Reservation System. The various objects in the Booking, Train Route, Customer, Ticket, and Train Schedule page-interact over the course of the sequence, and user will not be able to access this page without verifying their identity. This is the UML sequence diagram of Railway Reservation System which shows the interaction between the objects of Customer, Ticket, Train Route, Train Schedule, Booking. The instance of class objects involved in this UML Sequence Diagram of Railway Reservation System are as follows:

Customer Object

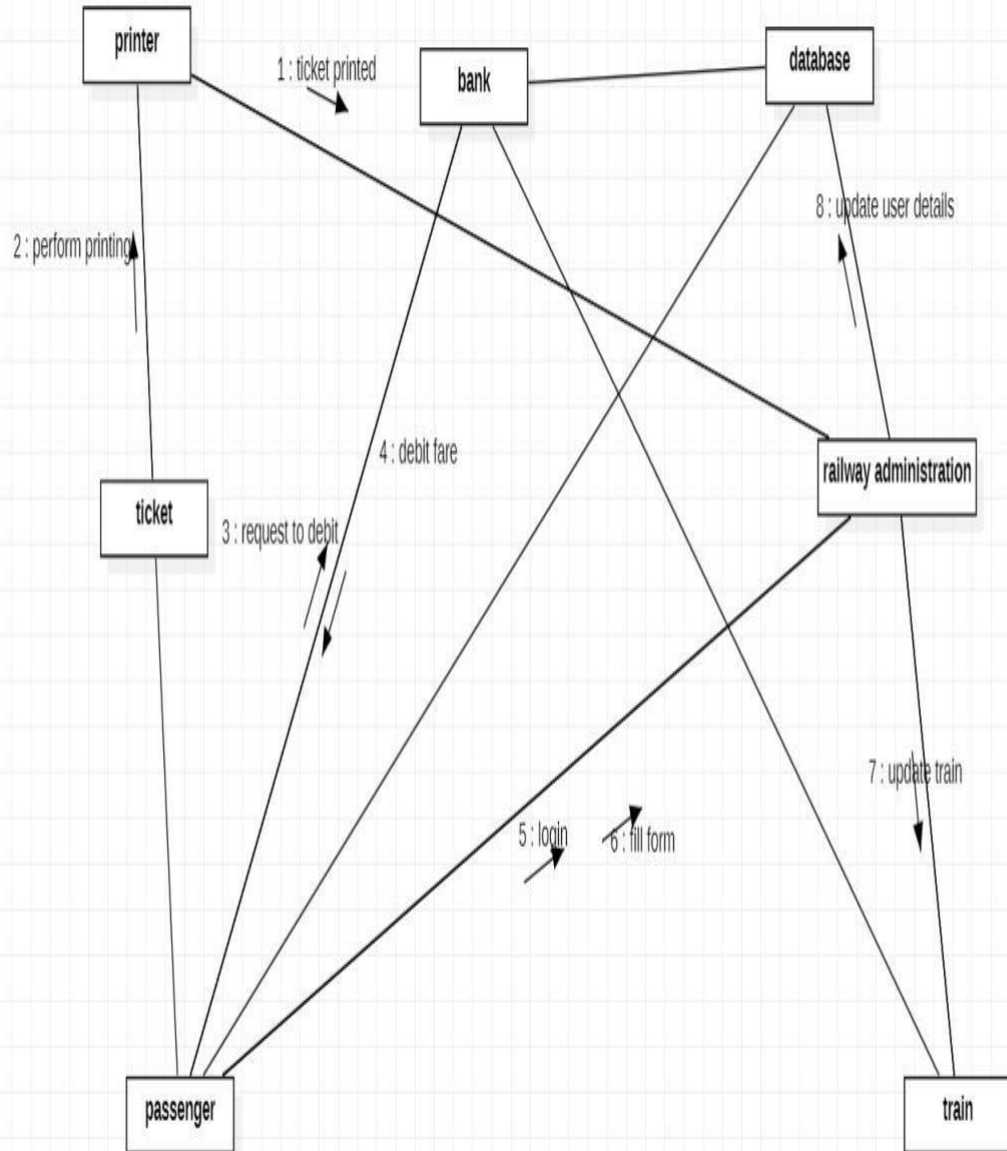
Ticket Object

Train Route Object

Train Schedule Object

Booking Object

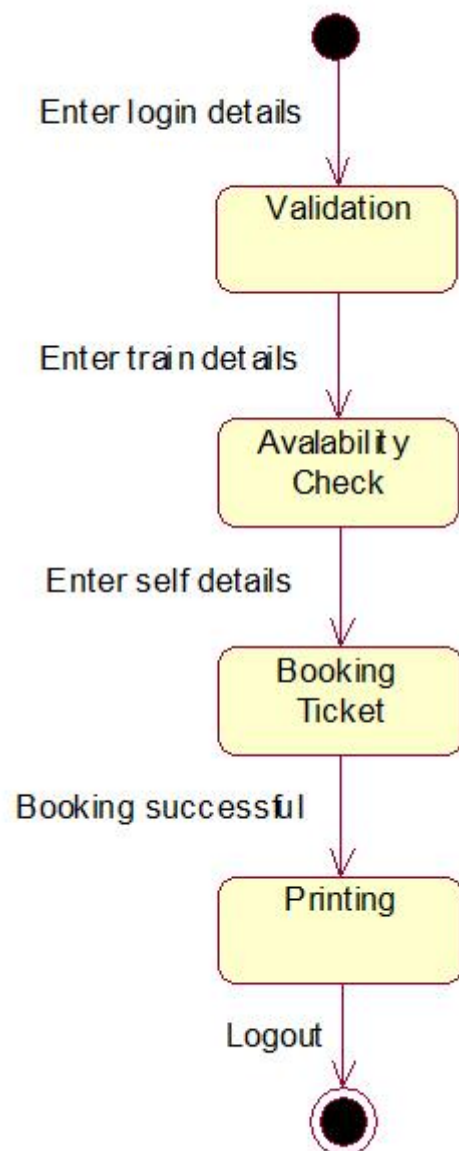
COMMUNICATION DIAGRAM



EXPLANATION:

The uml communication diagram has the link between passenger and train it means the passenger will login into the system and the administrator will check the details is correct if it is correct It will continue the process and the payment will be ask by the administrator and the user will redirect to the bank and the bank will asks the user for the payment type and the amount of fare will be debit from the users account and the the railway administration will ask the user to print ticket if the user want to cancel the ticket he should login with the user id and he should cancel the specific train he want to cancel not only this the railway administration will update the details of the train whether it is running or cancelled if the train has cancelled the railway team will refund the amount to authorized user to his bank account

STATE CHART DIAGRAM:



EXPLANATION:

This is the Activity UML diagram of Railway Reservation System which shows the flows between the activity of Ticket, Customer, Payment, Booking. Train Schedule. The main activity involved in this UML Activity Diagram of Railway Reservation System are as follows:

Ticket Activity

Customer Activity

Payment Activity

Booking Activity

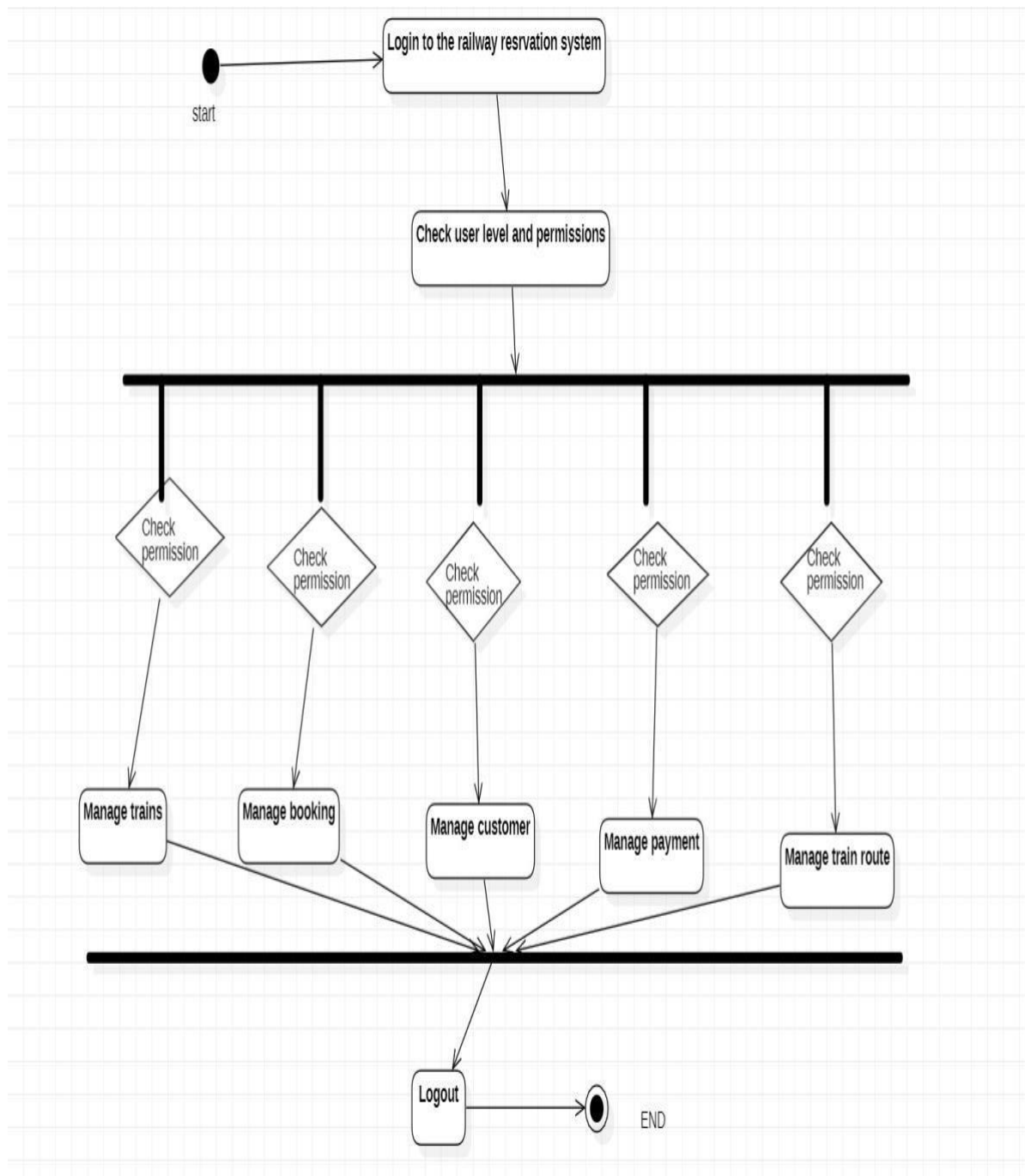
Train Schedule Activity

Features Of The Activity UML Diagram Of Railway Reservation System

Admin User can search Ticket, view description of a selected Ticket, add Ticket, update Ticket and delete Ticket.

Its shows the activity flow of editing, adding and updating of Customer

ACTIVITY DIAGRAM



EXPLANATION :

This is the Activity UML diagram of Railway Reservation System which shows the flows between the activity of Ticket, Customer, Payment, Booking, Train Schedule. The main activity involved in this UML Activity Diagram of Railway Reservation System are as follows:

Ticket Activity

Customer Activity

Payment Activity

Booking Activity

Train Schedule Activity

Features Of The Activity UML Diagram Of Railway Reservation System

Admin User can search Ticket, view description of a selected Ticket, add Ticket, update Ticket and delete Ticket.

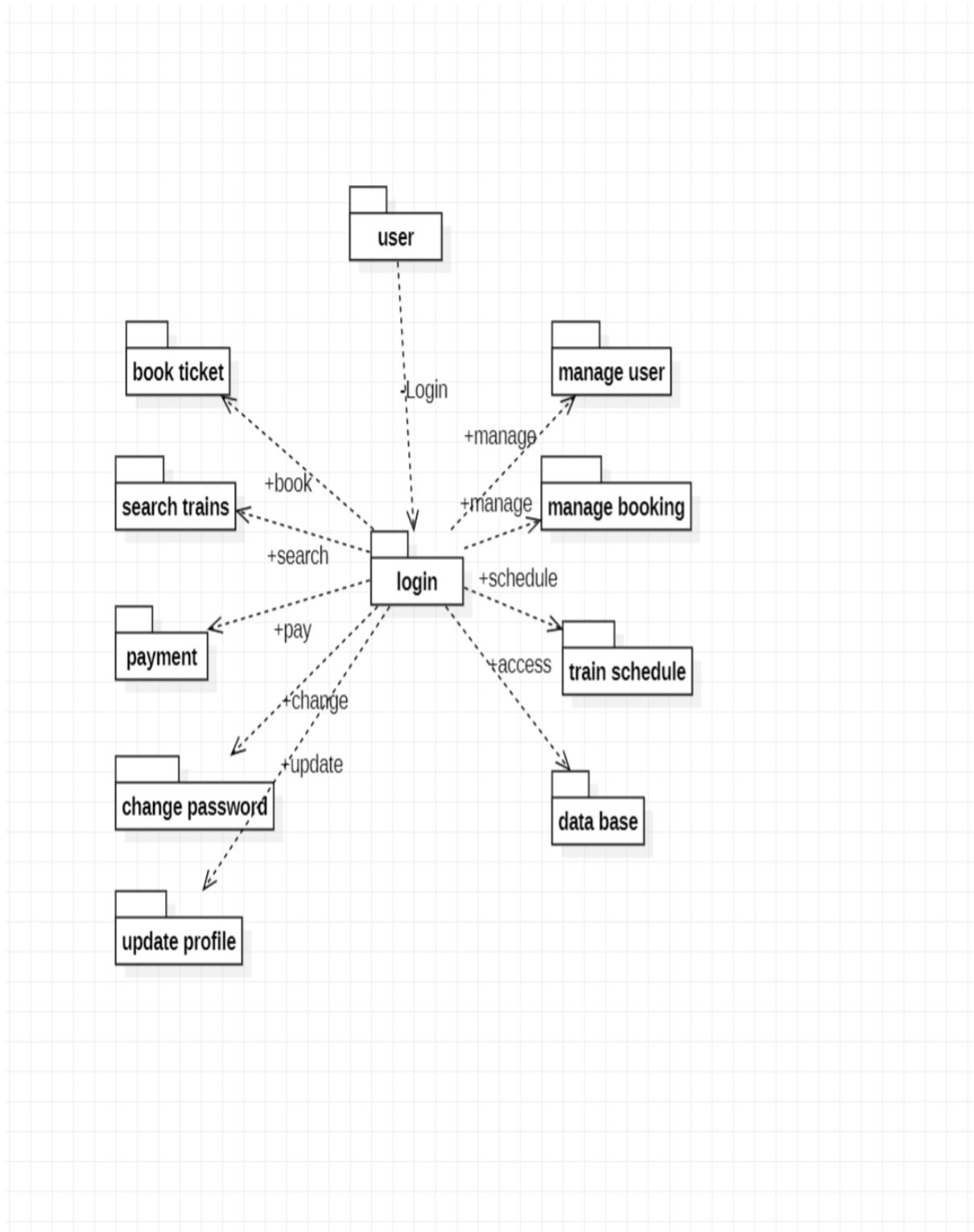
Its shows the activity flow of editing, adding and updating of Customer

User will be able to search and generate report of Payment, Booking, Train Schedule

All objects such as (Ticket, Customer, Train Schedule) are interlinked Its shows the full description and flow of Ticket,

Booking, Train Schedule, Payment, Customer

Package Diagram



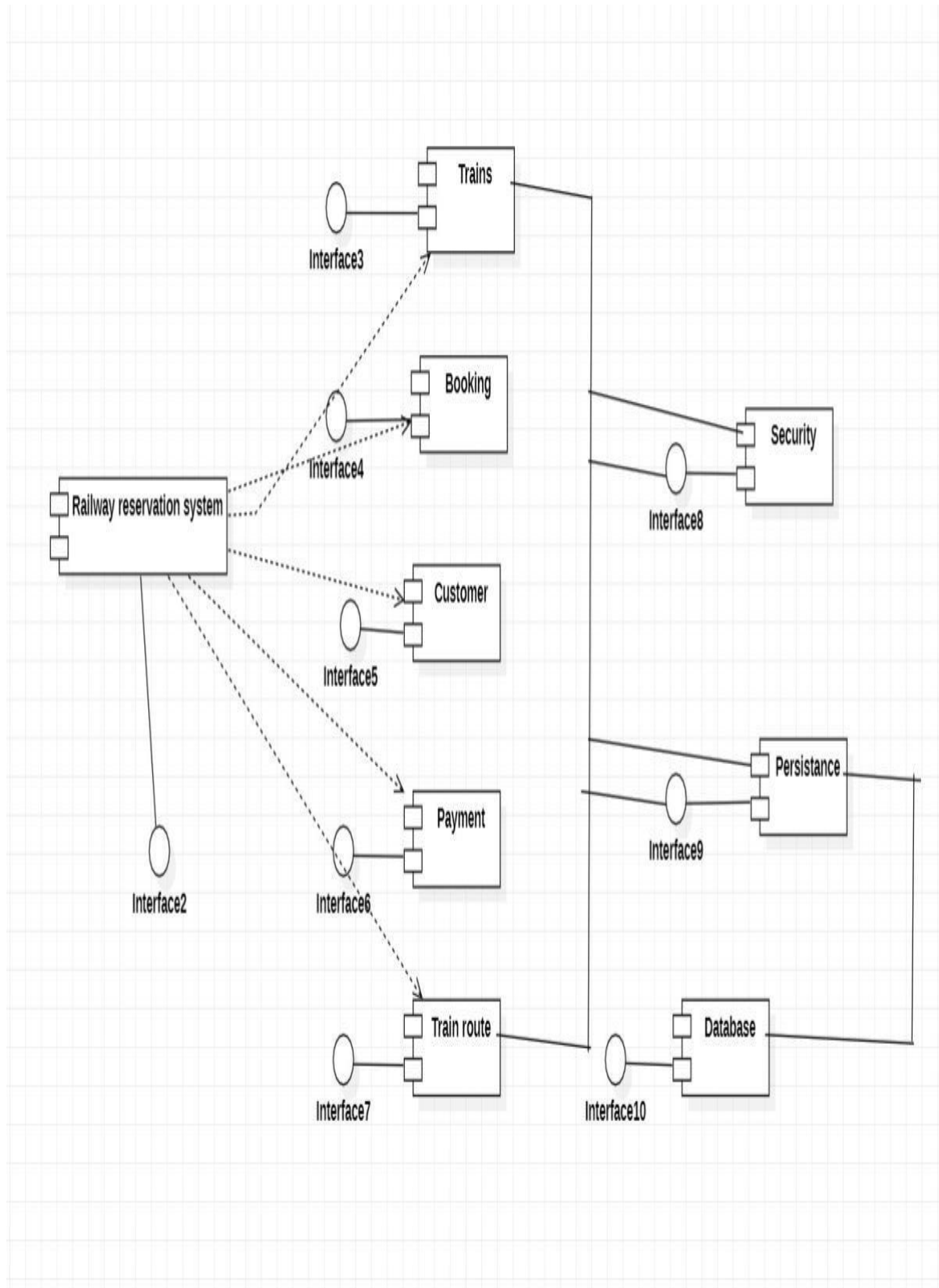
Explanation :

- Package diagrams are used to structure high level system elements. Packages are used for organizing large system which contains diagrams, documents and other key deliverables.
- Package diagrams are used to structure high level system elements. Packages are used for organizing large system which contains diagrams, documents and other key deliverables.
- Packages are depicted as file folders and can be used on any of the UML diagrams.

Components of package diagrams are:

- USER
- Book tickets
- Search trains
- Payment
- Change password
- Update profile
- Manage user
- Manage booking
- Train schedule
- Data base

COMPONENT DIAGRAM



EXPLANATION:

This is a Component diagram of Railway Reservation System which shows components, provided and required interfaces, ports, and relationships between the Train Schedule, Ticket, Booking, Customer and Payment. This type of diagram is used in Component-Based Development (CBD) to describe systems with

Service-Oriented Architecture (SOA). Railway Reservation System UML component diagram, describes the organization and wiring of the physical components in a system.

Components of UML Component Diagram of Railway Reservation System:

- Train Schedule Component

Ticket Component

Booking Component

- Customer Component

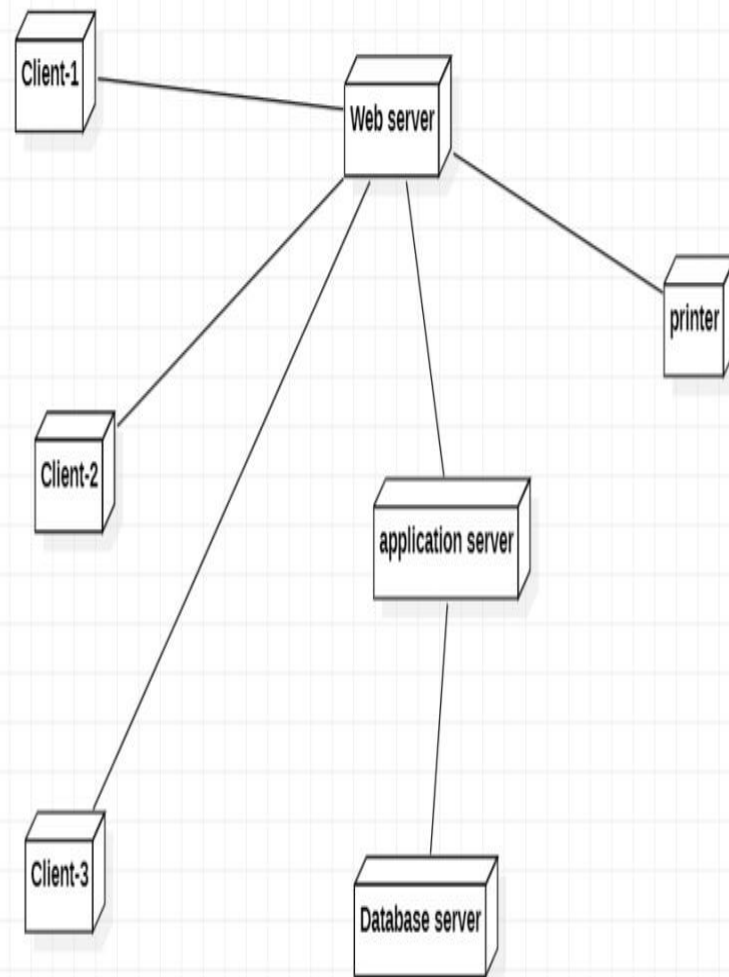
Payment Component

Features of Railway Reservation System Component Diagram:

. You can show the models the components of Railway Reservation System

Model the database schema of Railway Reservation System
Model the executables of an application of Railway Reservation System
Model the system's source code of Railway Reservation System

DEPLOYMENT DIAGRAM



EXPLANATION :

The railway reservation system UML deployment diagram explains the sketch of the relationship between software and hardware. These hardware and software are labeled to clarify their part in the system's operation. They were represented by nodes and the connections were represented by labeled arrows. The deployment diagram shows the scenario when the system is deployed. It has 6 nodes represented with boxes and relationship connections. The nodes are the main device (work station), the reservation system, information system, ISP, web server, and the Mobiles/PC. For the connection, all of the software was connected to ISP which enables it to pass data to the webserver. The main device uses the private network to secure all the data in the system. Other nodes need to have an internet connection to access the reservation and information system through URLs

CONCLUSION:

In our project Railway reservation system we have stored all the information about the Trains scheduled and the users booking tickets and even status of trains, seats etc. This data base is helpful for the applications which facilitate passengers to book the train tickets and check the details of trains and their status from their place itself it avoids inconveniences of going to railway station for each and every query they get. We had considered the most important requirements only, many more features and details can be added to our project in order to obtain even more user friendly applications. These applications are already in progress and in future they can be upgraded and may become part of amazing technology.

REFERENCE :

- **GOOGLE**
- **JAVATPOINT**
- **GEEKSFORGEEKS**
- **PROGRAMIZ**

CODESCOPE