RAILWAY RESERVATION SYSTEM

PROJECT REPORT

Submitted by

D.SADWIKA REDDY [RA2211003011072]
V.YASWANTH [RA2211003011123]

Under the Guidance of

Dr . M.KANDAN

Assistant Professor, Department of Computing Technologies

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

MAY 2023

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Project Report titled **".RAILWAY RESERVATION SYSTEM."** is the bonafide work done by **D.Sadwika reddy [RA2211003011072], V.Yaswanth [RA221103011123]** who completed the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other work.

SIGNATURE:
Dr.M.Kandan
OODP – Course Faculty
Department of SCHOOL OF COMPUTING
SRMIST

SIGNATURE:
DR .M.Pushpalatha
Head of the Department
Department of SCHOOL OF COMPUTING
SRMIST

TABLE OF CONTENTS

1. PROBLEM STATEMENT

You have been tasked with developing a Railway Reservation System in C++ that will allow users to book train tickets, check the availability of seats, and display information about the trains and the passengers.

The system should support the following features:
Add new trains to the system, including the train number, name, source, destination, and the number of seats available.

You have been tasked with developing a Railway Reservation System in C++ that will allow users to book train tickets, check the availability of seats, and display information about the trains and the passengers.

The system should support the following features:
- Add new trains to the system, including the train number, name, source, destination, and the number of seats available.

1.1.1 EXISTING SYSTEM

Add new trains to the system, including the train number,
name, source, destination, and the number of seats available.

1.2 OBJECTIVE OF THE PROJECT

Booking procedures are simple and user friendly. The process of booking an E-Ticket is easy and convenient. Registration on the website is free

2.MODULES OF THE PROJECT

1.Train Module

2.Passenger Module

3.Error Handling Module

4. Main Modu

2. 1 Train Module

This module is responsible for adding new trains to the system and displaying details of all available trains. It includes functions to add new trains, display train details, and validate train numbers

2.2 Passenger Module

This module is responsible for allowing passengers to book tickets by selecting a train and seat number. It includes functions to book tickets, display details of all booked tickets, and validate seat and train numbers.

2.3 Error Handling Module

This module is responsible for ensuring that invalid user input is not accepted. It includes functions to validate seat and train numbers and returns error messages if necessary.

2.4 Main Module

This module is the main driver of the system and is responsible for calling functions from the other modules as needed. It includes the main function that runs the program and displays a menu of options for the user to choose from

3. UML DIAGRAMS

3.1 SOFTWARE REQUIREMENTS/ HARDWARE REQUIREMENTS

3.1.1 Software Requirements

Operating System : windows 10

Frontend Languages : C++

Backend Languages : MS Access/SQL

3.1.2 Hardware Requirements

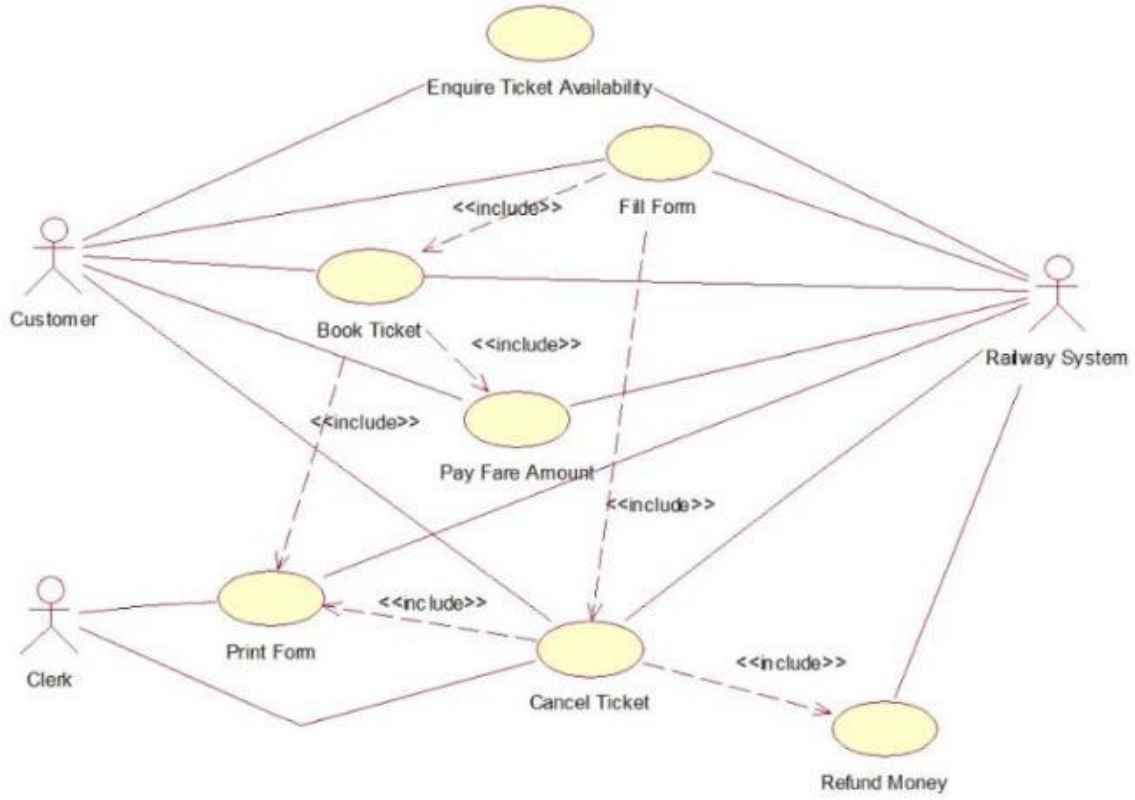
Processor : i5 and 1.90 GHz (Speed)

RAM : 1GB

Hard Disk : 512 MB

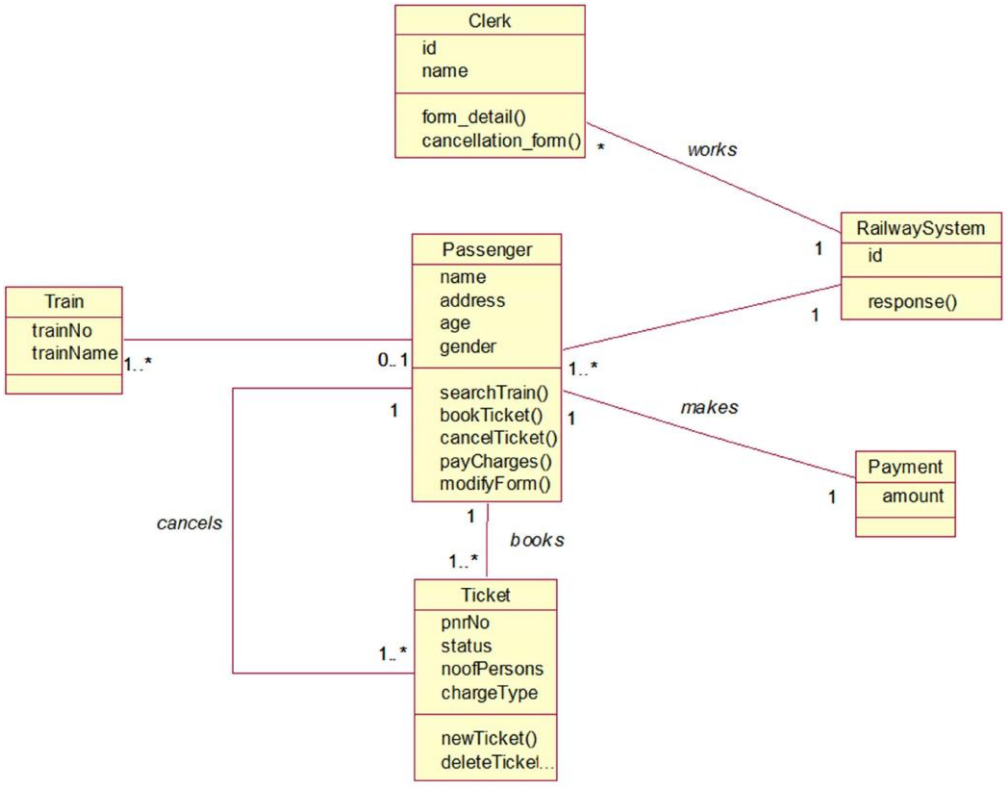
3.2 UML DIAGRAMS

a. Usecase Diagarm



Use cases of passengers are login, ticket availability, Filling the form, Book ticket, Canceling ticket, and Refund money.
Railway Reservation System: Use cases of the Railway Reservation System are login, ticket availability, Fill the form, Book ticket, Cancel ticket, and Refunding money.
Admin: use cases of Admin are Print ticket, refund money. Admin also controls the whole Railway Reservation System in different cases

b. Class Diagram

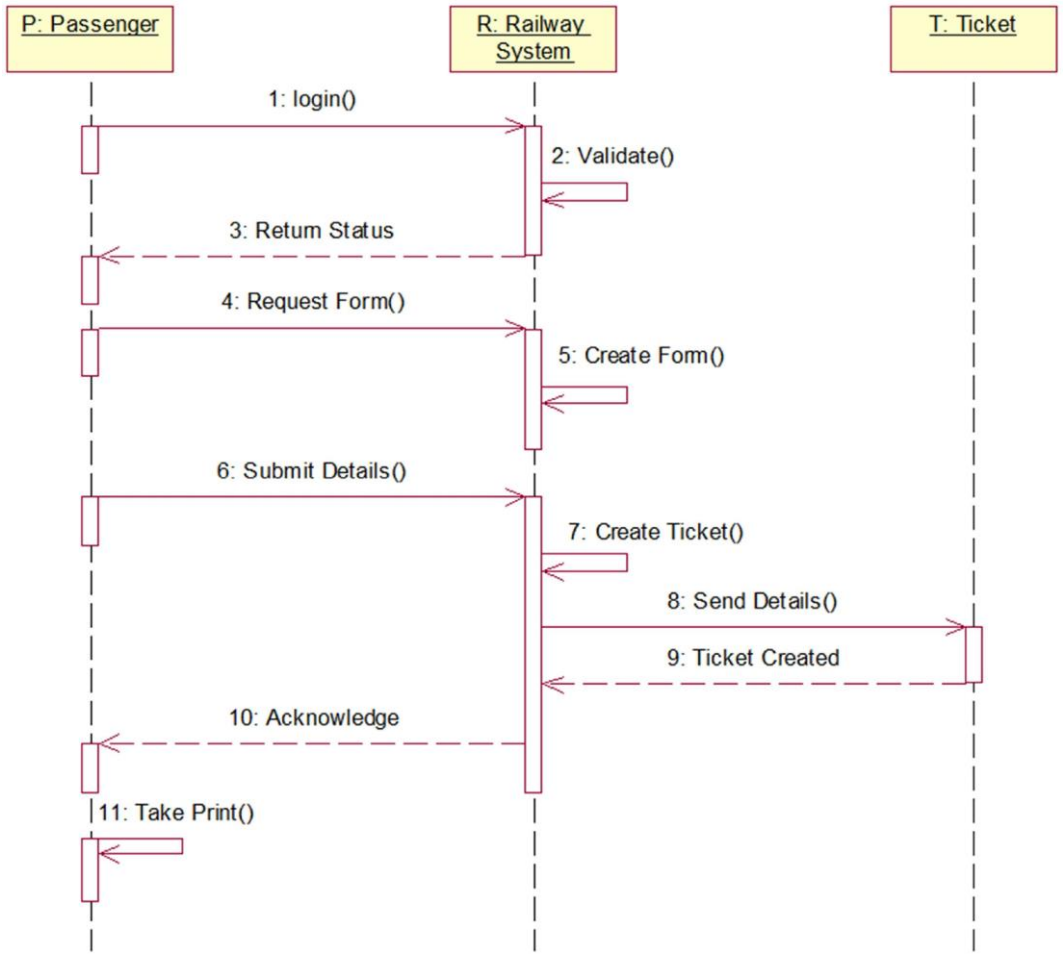


These diagrams describe the operation and attributes of a class with imposed constraints in the system. In this article the classes to be considered are ‘payment’, ‘train’, ‘passenger’, ‘ticket’, ‘railway reservation system’, ‘admin’

BEHAVIOUR DIAGRAM

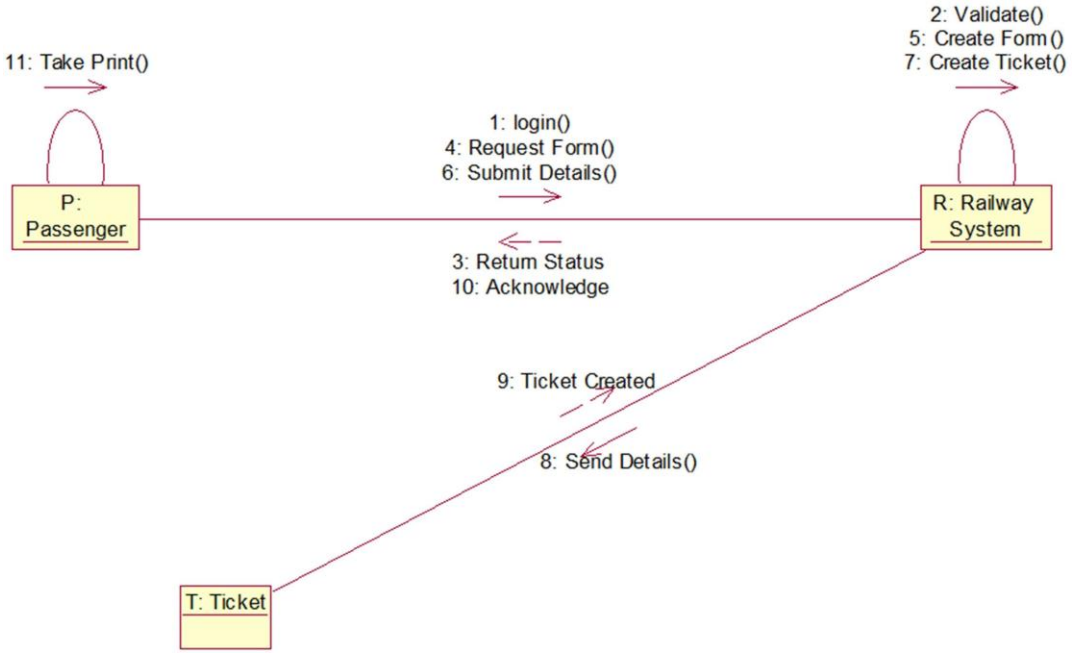
Interaction Diagram (Sequence Diagram and Collaboration Diagram)

c.sequence diagram



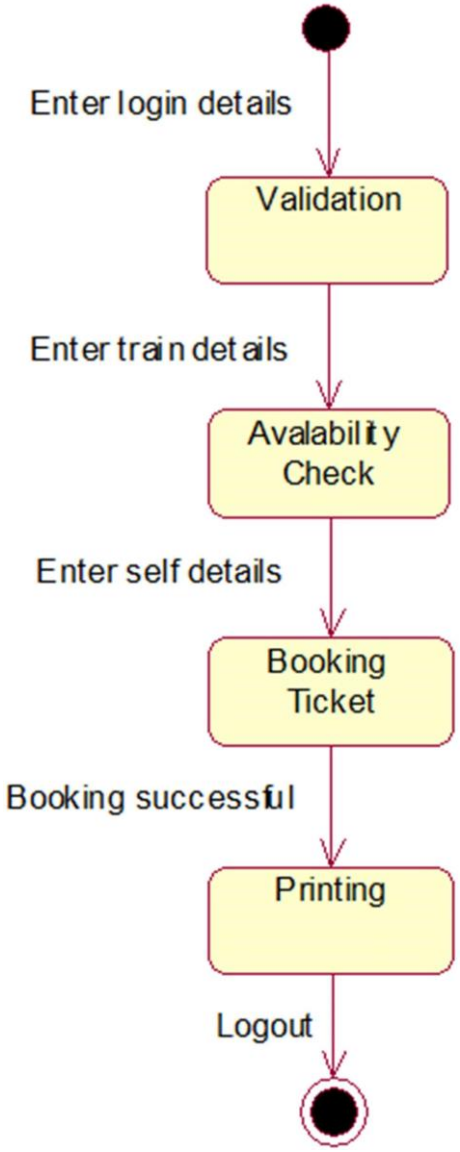
This diagram shows how and in which order a group of objects works together in a system. This is an interactive diagram and this is mostly used by software developers.

d.collaboration diagram



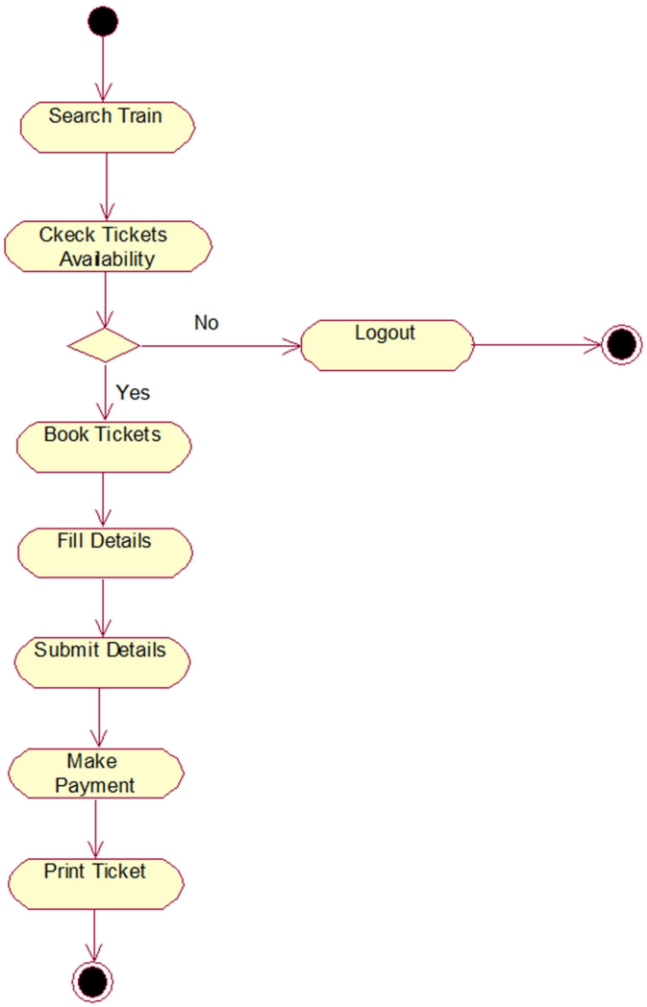
A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). Developers can use these diagrams to portray the dynamic behavior of a particular use case and define the role of each object.

e.state chart diagram



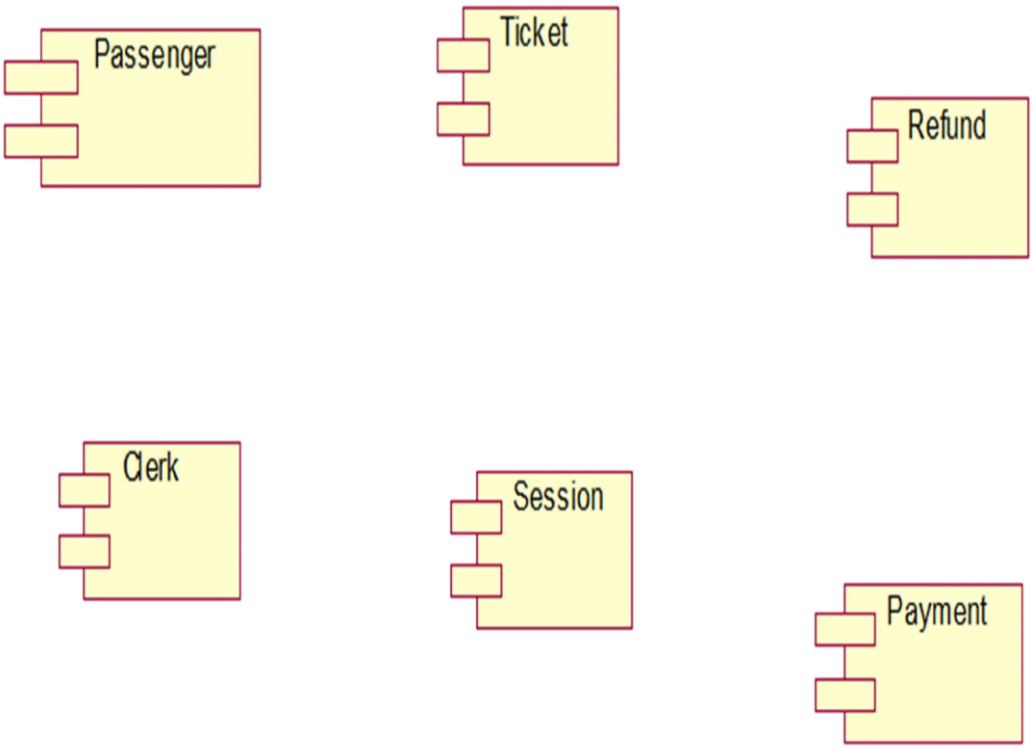
A state diagram (also known as a state machine or statechart diagram) is an illustration of all the possible behavioral states a software system component may exhibit and the various state changes it's predicted to undergo over the course of its operations.

f..activity diagram



This diagram shows the flow of processes from one to another activity. Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

g.package diagram

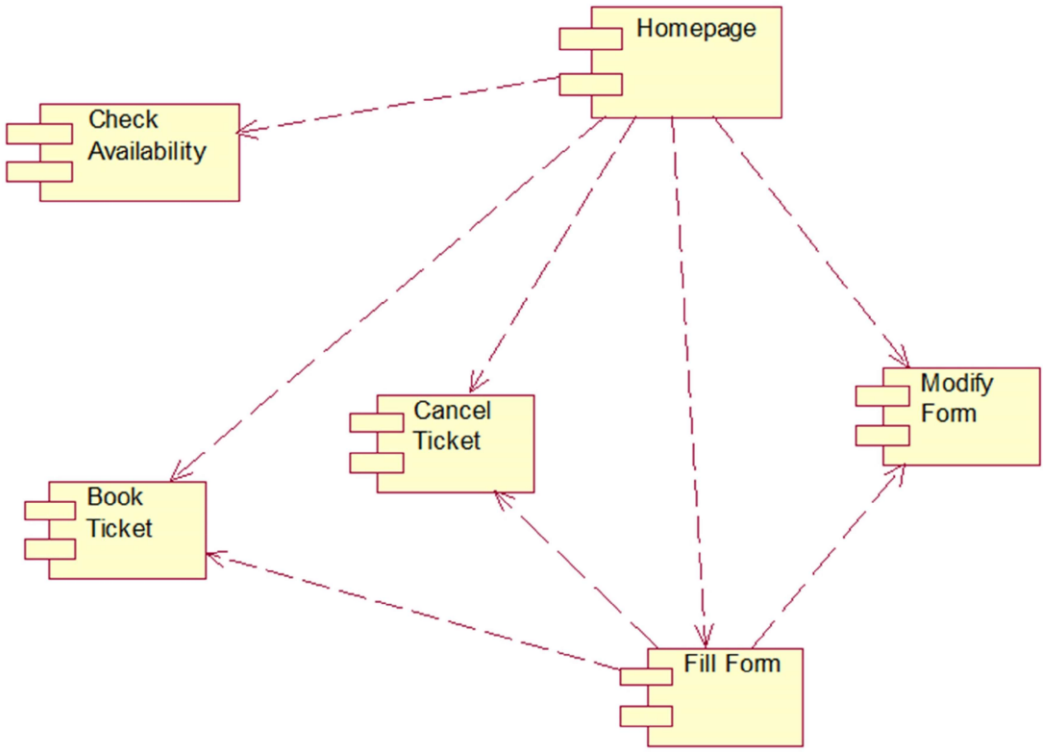


`

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages.

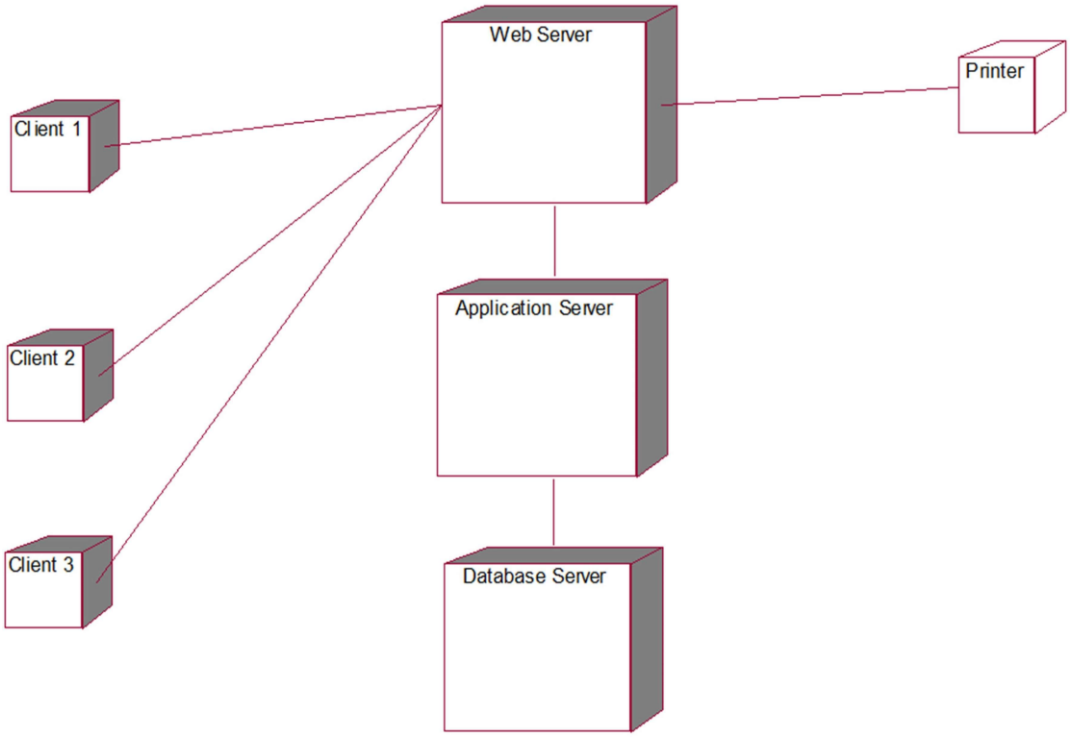
IMPLEMENTATION DIAGRAM

h..component diagram



This is a Component diagram of Railway Reservation System which shows components, provided and re- quired interfaces, ports, and relationships between the Train Schedule, Ticket, Booking, Customer and Payment.

i.deployment diagram



A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them

4. CODE/OUTPUT SCREENSHOTS

4.1 SOURCE CODE

```
include <iostream>
 #include <iomanip>
 #include <vector>
 #include <string> using namespace std;

  // Constants
  const int MAX_SEATS = 50;

  // Structures
  struct Passenger {
      string name;
      int age;
      string address;
      string contactNumber;
      int trainNumber;
      int seatNumber;
  };

  struct Train {
      int number;
      string name;
      string source;
      string destination;
      int seatsAvailable;
  };

  // Global Variables
  vector<Train> trains;
  vector<Passenger> passengers;

  // Function Prototypes
  void displayMenu();
  void addTrain();
  void displayTrains();
  void bookTicket();
  void displayPassengers();
  bool isValidTrainNumber(int);
  bool isValidSeatNumber(int, int);
```

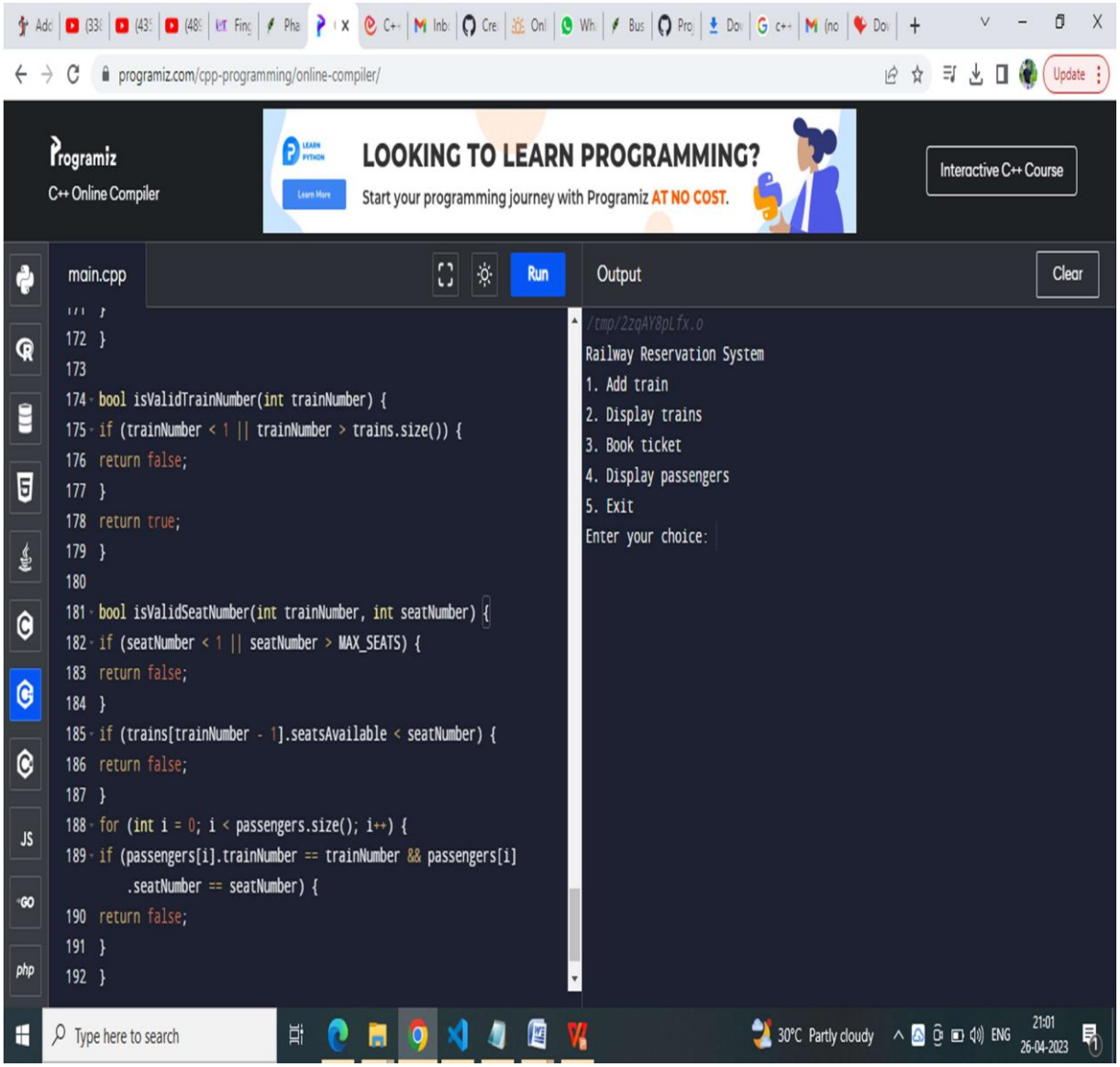
```
// Main Function
int main() {
    int choice;
    do {
        displayMenu();
        cin >> choice;
        switch (choice) {
            case 1:
                addTrain();
                break;
            case 2:
                displayTrains();
                break;
            case 3:
                bookTicket();
                break;
            case 4:
                displayPassengers();
                break;
            case 5:
                cout << "Exiting program." << endl;
                break;
            default:
                cout << "Invalid choice. Please try again." << endl;
        }
    } while (choice != 5);
    return 0;
}

// Function Definitions
void displayMenu() {
    cout << "Railway Reservation System" << endl;
    cout << "1. Add train" << endl;
    cout << "2. Display trains" << endl;
    cout << "3. Book ticket" << endl;
    cout << "4. Display passengers" << endl;
    cout << "5. Exit" << endl;
    cout << "Enter your choice: ";
}

void addTrain() {
    Train train;
    cout << "Enter train number: ";
```

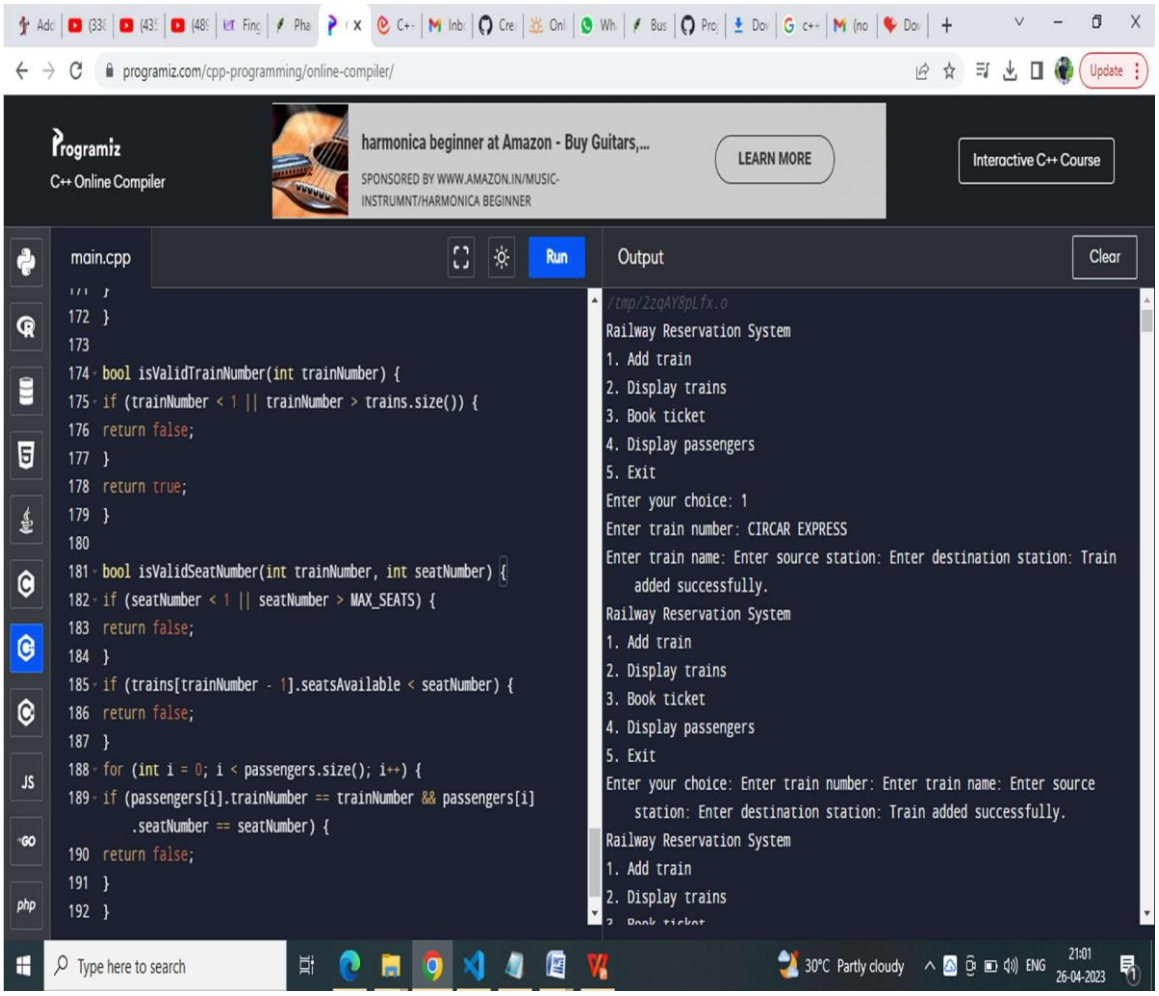
```
    cin >> train.number;
    cout << "Enter train name: ";
    cin.ignore();
    getline(cin, train.name);
    cout << "Enter source station: ";
    getline(cin, train.source);
    cout << "Enter destination station: ";
    getline(cin, train.destination);
    train.seatsAvailable = MAX_SEATS;
    trains.push_back(train);
    cout << "Train added successfully." << endl;
}

void displayTrains() {
    if (trains.empty()) {
        cout << "No trains available." << endl;
        return;
    }
    cout << left << setw(10) << "Number"
         << left << setw(20) << "Name"
         << left << setw(20) << "Source"
         << left << setw(20) << "Destination"
         << left << setw(10) << "Seats"
         << endl;
    for (int i = 0; i < trains.size(); i++) {
        cout << left << setw(10) << trains[i].number
             << left << setw(20) << trains[i].name
             << left << setw(20) << trains[i].source
             << left << setw(20) << trains[i].destination
             << left << setw(10) << trains[i].seatsAvailable
             << endl;
    }
}

void bookTicket() {
    if (trains.empty()) {
        cout << "No trains available to book tickets." << endl;
        return;
    }
    Passenger passenger;
    cout << "Enter passenger name: ";
    cin.ignore();
    getline(cin, passenger.name);
    cout << "Enter passenger age: ";
```

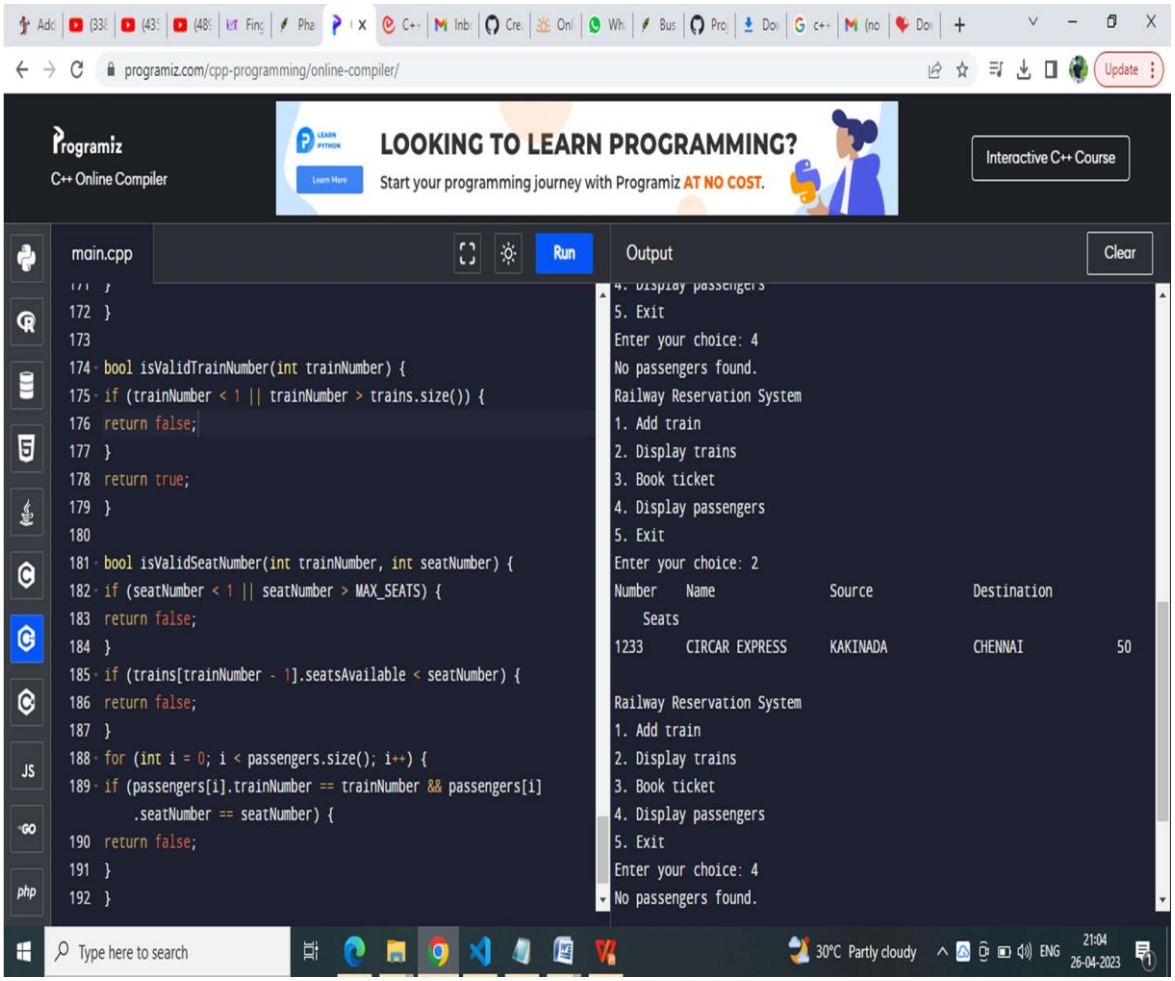
```
    cin >> passenger.age;
cout << "Enter passenger address: ";
cin.ignore();
getline(cin, passenger.address);
cout << "Enter passenger contact number: ";
getline(cin, passenger.contactNumber);
cout << "Enter train number: ";
cin >> passenger.trainNumber;
if (!isValidTrainNumber(passenger.trainNumber)) {
    cout << "Invalid train number." << endl;
    return;
}
cout << "Enter seat number: ";
cin >> passenger.seatNumber;
if (!isValidSeatNumber(passenger.trainNumber, passenger.seatNumber)) {
    cout << "Invalid seat number." << endl;
    return;
}
trains[passenger.trainNumber - 1].seatsAvailable--;
passengers.push_back(passenger);
cout << "Ticket booked successfully." << endl;
}

void displayPassengers() {
if (passengers.empty()) {
cout << "No passengers found." << endl;
return;
}
cout << left << setw(20) << "Name"
<< left << setw(10) << "Age"
<< left << setw(30) << "Address"
<< left << setw(20) << "Contact Number"
<< left << setw(10) << "Train"
<< left << setw(10) << "Seat"
<< endl;
for (int i = 0; i < passengers.size(); i++) {
cout << left << setw(20) << passengers[i].name
<< left << setw(10) << passengers[i].age
<< left << setw(30) << passengers[i].address
<< left << setw(20) << passengers[i].contactNumber
<< left << setw(10) << passengers[i].trainNumber
<< left << setw(10) << passengers[i].seatNumber
<< endl;
}
```

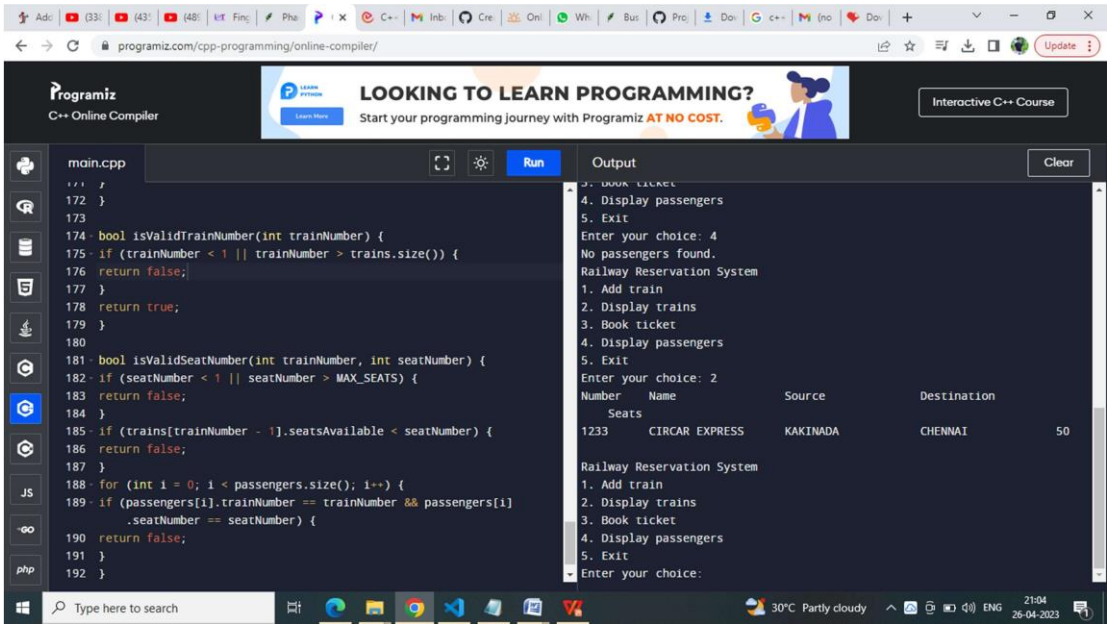
```
}

bool isValidTrainNumber(int trainNumber) {
if (trainNumber < 1 || trainNumber > trains.size()) {
return false;
}
return true;
}

bool isValidSeatNumber(int trainNumber, int seatNumber) {
if (seatNumber < 1 || seatNumber > MAX_SEATS) {
return false;
}
if (trains[trainNumber - 1].seatsAvailable < seatNumber) {
return false;
}
for (int i = 0; i < passengers.size(); i++) {
if (passengers[i].trainNumber == trainNumber && passengers[i].seatNumber ==
seatNumber) {
return false;
}
}
return true;
}
```

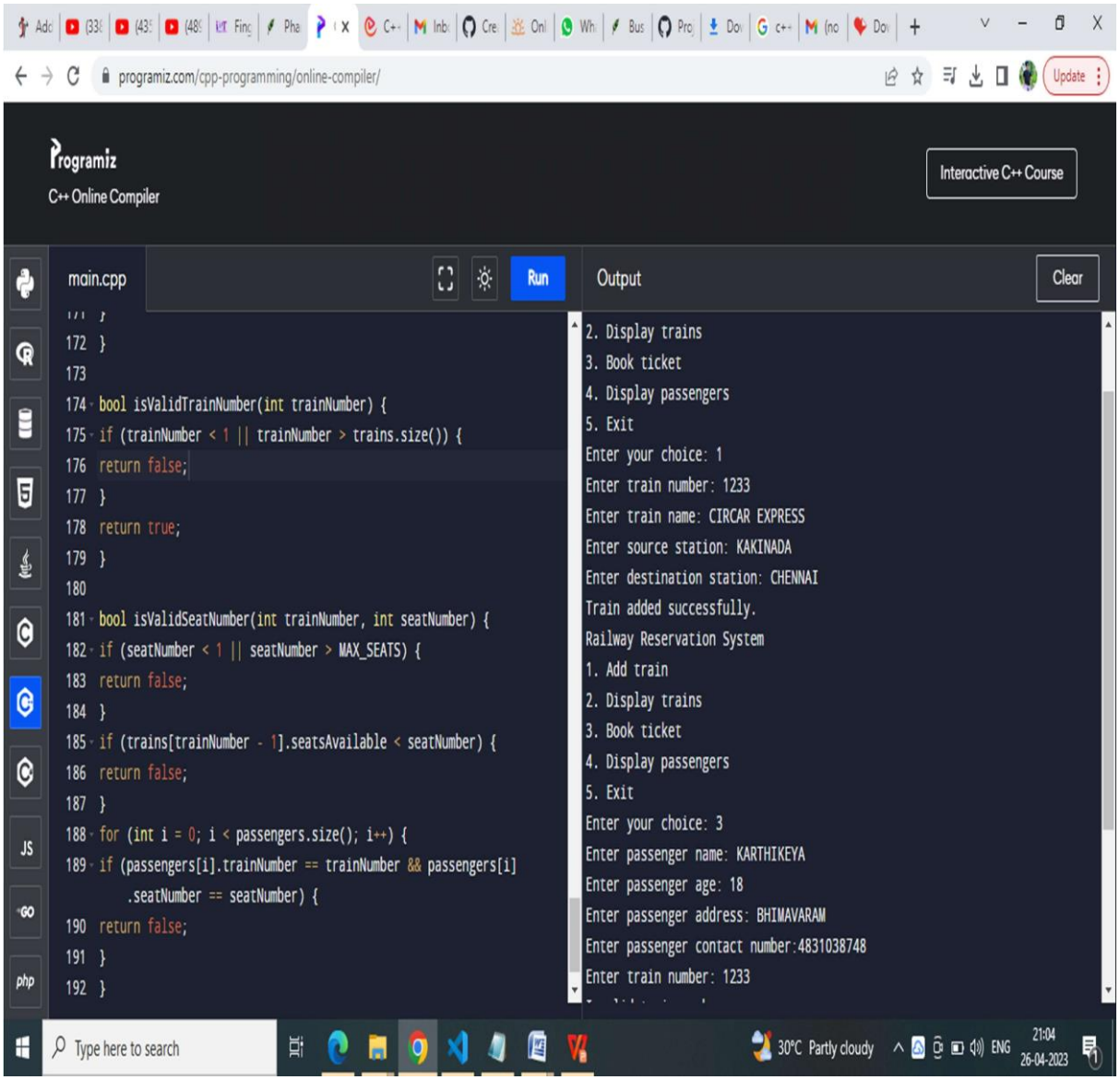
4.2 SCREENSHOTS




programiz.com/cpp-programming/online-compiler/
Programiz
C++ Online Compiler
harmonica beginner at Amazon - Buy Guitars,...
SPONSORED BY WWW.AMAZON.IN/MUSIC-INSTRUMNT/HARMONICA BEGINNER
LEARN MORE
Interactive C++ Course
main.cpp
Run
Output
Clear
172 }
173
174 bool isValidTrainNumber(int trainNumber) {
175 if (trainNumber < 1 || trainNumber > trains.size()) {
176 return false;
177 }
178 return true;
179 }
180
181 bool isValidSeatNumber(int trainNumber, int seatNumber) {
182 if (seatNumber < 1 || seatNumber > MAX_SEATS) {
183 return false;
184 }
185 if (trains[trainNumber - 1].seatsAvailable < seatNumber) {
186 return false;
187 }
188 for (int i = 0; i < passengers.size(); i++) {
189 if (passengers[i].trainNumber == trainNumber && passengers[i]
.seatNumber == seatNumber) {
190 return false;
191 }
192 }
/tmp/2zqAY8pLfx.o
Railway Reservation System
1. Add train
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice: 1
Enter train number: CIRCAR EXPRESS
Enter train name: Enter source station: Enter destination station: Train added successfully.
Railway Reservation System
1. Add train
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice: Enter train number: Enter train name: Enter source station: Enter destination station: Train added successfully.
Railway Reservation System
1. Add train
2. Display trains
Type here to search
30°C Partly cloudy
ENG
21:01
26-04-2023


programiz.com/cpp-programming/online-compiler/
Programiz
C++ Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Interactive C++ Course
main.cpp
Run
Output
Clear
172 }
173
174 bool isValidTrainNumber(int trainNumber) {
175 if (trainNumber < 1 || trainNumber > trains.size()) {
176 return false;
177 }
178 return true;
179 }
180
181 bool isValidSeatNumber(int trainNumber, int seatNumber) {
182 if (seatNumber < 1 || seatNumber > MAX_SEATS) {
183 return false;
184 }
185 if (trains[trainNumber - 1].seatsAvailable < seatNumber) {
186 return false;
187 }
188 for (int i = 0; i < passengers.size(); i++) {
189 if (passengers[i].trainNumber == trainNumber && passengers[i]
.seatNumber == seatNumber) {
190 return false;
191 }
192 }
5. Exit
Enter your choice: 4
No passengers found.
Railway Reservation System
1. Add train
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice: 2
Number Name Source Destination
Seats
1233 CIRCAR EXPRESS KAKINADA CHENNAI 50
Railway Reservation System
1. Add train
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice: 4
No passengers found.
Type here to search
30°C Partly cloudy
ENG
21:04
26-04-2023


programiz.com/cpp-programming/online-compiler/
Programiz
C++ Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Interactive C++ Course
main.cpp
Run
Output
Clear
}
bool isValidTrainNumber(int trainNumber) {
if (trainNumber < 1 || trainNumber > trains.size()) {
return false;
}
return true;
}
bool isValidSeatNumber(int trainNumber, int seatNumber) {
if (seatNumber < 1 || seatNumber > MAX_SEATS) {
return false;
}
if (trains[trainNumber - 1].seatsAvailable < seatNumber) {
return false;
}
for (int i = 0; i < passengers.size(); i++) {
if (passengers[i].trainNumber == trainNumber && passengers[i]
.seatNumber == seatNumber) {
return false;
}
}
4. Display passengers
5. Exit
Enter your choice: 4
No passengers found.
Railway Reservation System
1. Add train
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice: 2
Number Name Source Destination
Seats
1233 CIRCAR EXPRESS KAKINADA CHENNAI 50
Railway Reservation System
1. Add train
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice:
Type here to search
30°C Partly cloudy
ENG
21:04
26-04-2023


programiz.com/cpp-programming/online-compiler/
Update
Programiz
C++ Online Compiler
Interactive C++ Course
main.cpp
Run
Output
Clear
}
}
bool isValidTrainNumber(int trainNumber) {
if (trainNumber < 1 || trainNumber > trains.size()) {
return false;
}
return true;
}
bool isValidSeatNumber(int trainNumber, int seatNumber) {
if (seatNumber < 1 || seatNumber > MAX_SEATS) {
return false;
}
if (trains[trainNumber - 1].seatsAvailable < seatNumber) {
return false;
}
for (int i = 0; i < passengers.size(); i++) {
if (passengers[i].trainNumber == trainNumber && passengers[i]
.seatNumber == seatNumber) {
return false;
}
}
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice: 1
Enter train number: 1233
Enter train name: CIRCAR EXPRESS
Enter source station: KAKINADA
Enter destination station: CHENNAI
Train added successfully.
Railway Reservation System
1. Add train
2. Display trains
3. Book ticket
4. Display passengers
5. Exit
Enter your choice: 3
Enter passenger name: KARTHIKEYA
Enter passenger age: 18
Enter passenger address: BHIMAVARAM
Enter passenger contact number:4831038748
Enter train number: 1233
Type here to search
30°C Partly cloudy
ENG
21:04
26-04-2023

5. RESULTS AND CONCLUSION

5.1. RESULTS

THIS IS A PROJECT BASED ON RAILWAY
RESERVATION SYSTEM
BY THIS YOU CAN DO FOLLOWING FUNCTIONS:
1.BOOK TICKET
2.ADD TRAIN
3.DISPLAY TRAIN
4.DISPLAY PASSENGERS
5. EXIT

5.1.1 Results Comparison

For the Railway Reservation system we choose spiral model. The spiral model is a very beneficial and efficient model. It includes various activities, such as traffic planning, operation management, power supply & infrastructure management, maintenance & support, station control & communication network, rail-facility information management, and others.

5.1.2Application

Online ticket reservation can be done by using IRCTC website or IRCTC app. Computer reservation systems, or central reservation systems (CRS), are computerized systems used to store and retrieve information and conduct transactions related to air travel, hotels, car rental, or other activities.

5.2 CONCLUSION

THIS PROJECT MAKE US LEARN HOW TO
WRITE A CODE FOR A PARTICULAR PROBLEM STATEMENT AND WE HAVE
SUCCESFULY LEARNT TO DRAW DIFFERENT
UML DIAGRAMS

5.2.1 FUTURE ENHANCEMENT

Train travel is in a disadvantaged position in our world. After dominating transport for over 100 years, railways are no longer competitive with air travel. And if the fairness of this comparison concerns you, all you have to do is consider how expectations for purchasing railway tickets were influenced by the way we purchase airline tickets.

6. REFERENCES

1.The C++ Programming
Language:https://books.google.co.in/books/about/The_C
++_Programming_Language.html?id=PSUNAAAAQB
AJ&source=kp_book_description&redir_esc=y
2.github https://github.com/
3.C++ PRIMER BOOK
4. modern c++ design book