

**GOVERNMENT POLYTECHNIC, AHMEDABAD**  
**COMPUTER ENGINEERING DEPARTMENT**



**Affiliated  
To  
Gujarat Technological University, Ahmedabad**

**Micro project Report**  
**D. E. Second Year (Semester–IV)**

**Advanced Object-Oriented Programming  
(4340701)**



**Government Polytechnic, Ahmedabad**  
**Computer Engineering Department**

**CERTIFICATE**

This is to certify that

Sr. No.	Enrollment No.	Name
1	226170307235	Yadav Prajwal P.

Of **Fourth** semester of Diploma in Computer Engineering of Government Polytechnic, Ahmedabad has completed the Micro-Project satisfactorily in Subject **Introduction to Software Engineering(4340702)** for the academic year **20223-2024** asprescribed in the curriculum.

Lecturer,  
Computer Engg. Dept.,  
Government Polytechnic, Ahmedabad

HOD  
Computer Engg. Dept.,  
Government Polytechnic,  
Ahmedabad

## **RUBRICS FOR MICRO-PROJECT ASSESMENT**

Parameters	Allocated Marks	High	Medium	Low
<b>Problem Analysis and Solution(R1)</b>	8	Problem is Properly Analyzed and Solved	Problem is Properly Analyzed but Partially Solved	Problem is properly Analyzed but not Solved.
		8 Marks	5 Marks	2 Marks
<b>Viva Voce(R2)</b>	2	Student Answered All The Viva Voce Questions	Student Answered Only A Few Viva Voce Questions	Student Did Not Answer Any Viva Voce Questions
		2 Marks	1 Marks	0 Marks

## **INDEX**

1. INRODUCTION
2. SYSTEM OVERVIEW
3. REQUERMENTS
4. DESIGN
5. IMPLEMENTATION
6. TESTING

Enrollment Number	Student Name	Marks(R1)	Marks(R2)	Total Marks
226170307235	Yadav Prajwal P.			
Name and Sign of Faculty:				

# Airline Reservation System

## 1. Introduction

### Brief Overview of the Project

The Airline Reservation System is a simple console-based application developed in Java. It enables users to reserve and cancel seats on a flight, and view available seats. The system manages a fixed number of seats and allows operations such as booking and cancellation through a command-line interface.

### Purpose of the System

The primary purpose of the Airline Reservation System is to simulate a basic reservation system for an airline. It demonstrates the fundamental concepts of data management, user input handling, and basic operations such as reservation, cancellation, and seat availability checking. This system is designed for educational purposes to help understand the basic structure and functionality of reservation systems.

### Objectives

**Ease of Use:** Provide a simple and intuitive interface for users to interact with the system.

**Basic Functionality:** Implement essential functionalities such as seat reservation, cancellation, and viewing available seats.

**Data Management:** Efficiently manage the seat reservations and ensure data consistency.

**User Interaction:** Allow users to perform operations through a command-line interface, enhancing their understanding of basic Java programming concepts.

## 2. System Overview

### Description of the Airline Reservation System

The Airline Reservation System manages a fixed number of seats (10 seats) and allows users to reserve, cancel, and view the status of these seats. Each seat can be booked by a single passenger, and the system prevents double booking of the same seat.

### Features and Functionalities

**Reserve Seat:** Allows users to reserve a seat by entering their name.

**Cancel Reservation:** Allows users to cancel a reservation by providing the seat number.

**View Available Seats:** Displays a list of all unoccupied seats.

### User Roles

**User:** Can perform seat reservations, cancellations, and view available seats.

## 3. Requirements

### Functional Requirements

The system should allow users to reserve a seat by entering their name.

The system should allow users to cancel a reservation by providing the seat number.

The system should display all available seats when requested.

The system should prevent double booking of the same seat.

### Non-functional Requirements

**Performance:** The system should provide quick responses to user actions.

**Usability:** The system should have an intuitive command-line interface.

**Reliability:** The system should maintain accurate seat reservation data.

### User Interface Requirements

The interface should be text-based and run in a console window.

It should prompt users for necessary input and display appropriate messages.

## 4. Code

```
import java.util.Scanner;
class Passenger {
    private String name;
    private int seatNumber;

    public Passenger(String name, int seatNumber) {
        this.name = name;
        this.seatNumber = seatNumber;
    }

    public String getName() {
        return name;
    }

    public int getSeatNumber() {
        return seatNumber;
    }
}
public class AirlineReservationSystem {
    private static final int NUM_SEATS = 10;
    private Passenger[] seats;
    private Scanner scanner;

    public AirlineReservationSystem() {
        seats = new Passenger[NUM_SEATS];
        scanner = new Scanner(System.in);
    }

    public void displayMenu() {
        System.out.println("Welcome to the Airline Reservation System");
        System.out.println("1. Reserve Seat");
        System.out.println("2. Cancel Reservation");
        System.out.println("3. View Available Seats");
        System.out.println("4. Exit");
    }

    public void reserveSeat() {
        System.out.print("Enter passenger name: ");
        String name = scanner.nextLine();

        int seatNumber = findAvailableSeat();
        if (seatNumber != -1) {
            seats[seatNumber] = new Passenger(name, seatNumber);
            System.out.println("Seat reserved successfully. Your seat number is " +
seatNumber);
        } else {
            System.out.println("Sorry, all seats are occupied.");
        }
    }

    private int findAvailableSeat() {
        for (int i = 0; i < NUM_SEATS; i++) {
            if (seats[i] == null) {
                return i;
            }
        }
    }
}
```

```

        }
    }
    return -1; // No available seats
}

public void cancelReservation() {
    System.out.print("Enter seat number to cancel reservation: ");
    int seatNumber = scanner.nextInt();
    if (seatNumber >= 0 && seatNumber < NUM_SEATS && seats[seatNumber] != null) {
        seats[seatNumber] = null;
        System.out.println("Reservation cancelled successfully.");
    } else {
        System.out.println("Invalid seat number or no reservation found for that
seat.");
    }
}

public void viewAvailableSeats() {
    System.out.println("Available seats:");
    for (int i = 0; i < NUM_SEATS; i++) {
        if (seats[i] == null) {
            System.out.print(i + " ");
        }
    }
    System.out.println();
}

public static void main(String[] args) {
    AirlineReservationSystem reservationSystem = new AirlineReservationSystem();
    Scanner scanner = new Scanner(System.in);
    int choice;

    do {
        reservationSystem.displayMenu();
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        switch (choice) {
            case 1:
                reservationSystem.reserveSeat();
                break;
            case 2:
                reservationSystem.cancelReservation();
                break;
            case 3:
                reservationSystem.viewAvailableSeats();
                break;
            case 4:
                System.out.println("Thank you for using the Airline Reservation
System. Goodbye!");
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    } while (choice != 4);
}
}

```



## 5. Output :

```
java -cp /tmp/4BVsjgIzgE/AirlineReservationSystem
Welcome to the Airline Reservation System
1. Reserve Seat
2. Cancel Reservation
3. View Available Seats
4. Exit
Enter your choice: 1
Enter passenger name: Prajwal
Seat reserved successfully. Your seat number is 0
```

```
Welcome to the Airline Reservation System
1. Reserve Seat
2. Cancel Reservation
3. View Available Seats
4. Exit
Enter your choice: 3
Available seats:
2 3 4 5 6 7 8 9
```

```
Welcome to the Airline Reservation System
1. Reserve Seat
2. Cancel Reservation
3. View Available Seats
4. Exit
Enter your choice: 2
Enter seat number to cancel reservation: 1
Reservation cancelled successfully.
```

```
Welcome to the Airline Reservation System
1. Reserve Seat
2. Cancel Reservation
3. View Available Seats
4. Exit
Enter your choice: 4
Thank you for using the Airline Reservation System. Goodbye!

=== Code Execution Successful ===
```

## 6. Conclusion

### Summary of the Project

The Airline Reservation System successfully simulates a basic reservation system. It includes essential functionalities such as seat reservation, cancellation, and availability checking. The project serves as a practical example for understanding basic concepts in Java programming and data management.

# Thank You....!