Informatics Institute of Technology

School of Computing

Software Development II Coursework Report

Module : 4COSC010C.2: Software Development II (2023)

Date of submission : 25th March 2024

Student ID : 20230774 / w2053162

Student First Name : Nimsara

Student Surname : Mahith

Tutorial group (day, time, and tutor/s): Monday, 10.30 a.m-12.30 p.m, mr. Dilshad Ahamed

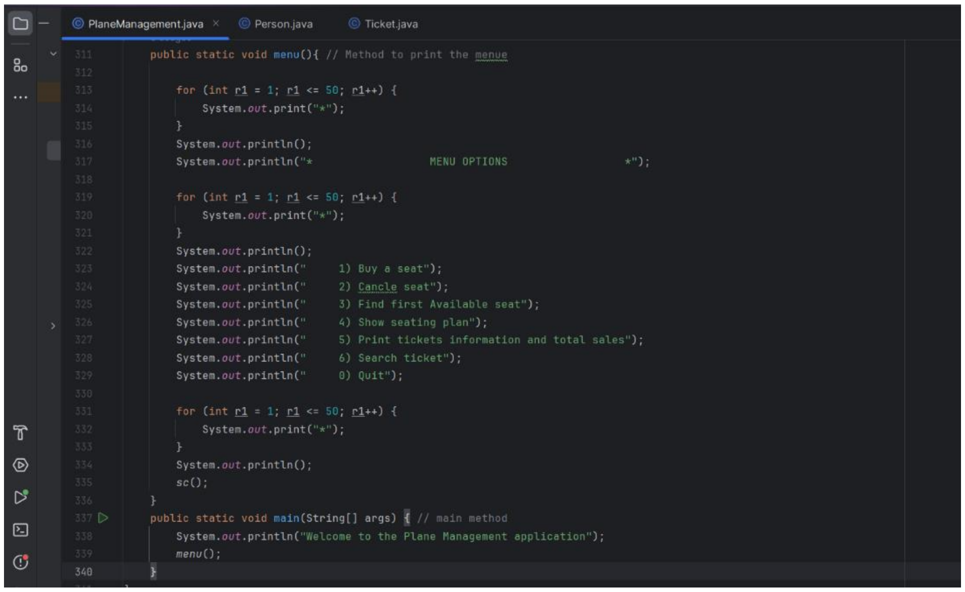
Name : K.V.A Nimsara Mahith

Student ID : 20230774/w2053162

Self-assessment form and test plan

1) Self-assessment form

Task	Self-assessment (selectone)	Comments
1	Fully implemented	Plane Management Project and class was created and the main method was created to print th welcome message
2	Fully implemented	Menu was printed and the method to select menu items was created



PlaneManagement.java   Person.java   Ticket.java

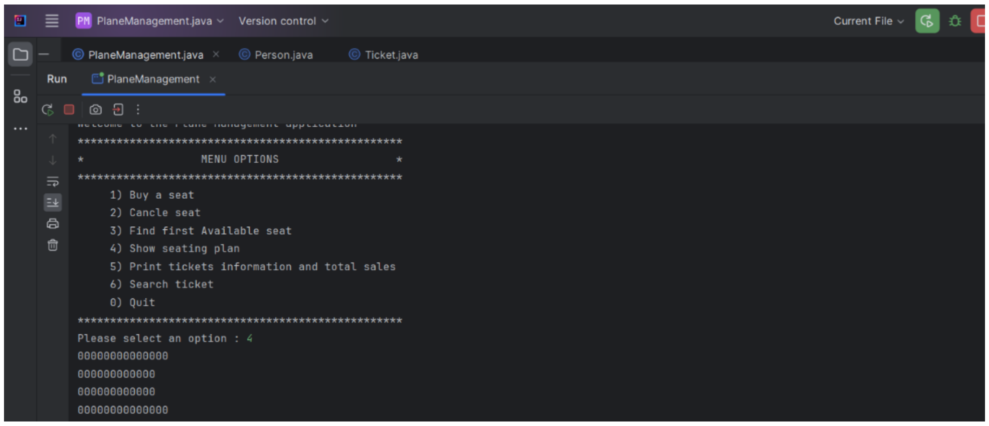
```java
    public static void menu(){ // Method to print the menue

        for (int r1 = 1; r1 <= 50; r1++) {
            System.out.print("*");
        }
        System.out.println();
        System.out.println("*                  MENU OPTIONS                  *");

        for (int r1 = 1; r1 <= 50; r1++) {
            System.out.print("*");
        }
        System.out.println();
        System.out.println("     1) Buy a seat");
        System.out.println("     2) Cancle seat");
        System.out.println("     3) Find first Available seat");
        System.out.println("     4) Show seating plan");
        System.out.println("     5) Print tickets information and total sales");
        System.out.println("     6) Search ticket");
        System.out.println("     0) Quit");

        for (int r1 = 1; r1 <= 50; r1++) {
            System.out.print("*");
        }
        System.out.println();
        sc();
    }
    public static void main(String[] args) { // main method
        System.out.println("Welcome to the Plane Management application");
        menu();
    }
```

3	Fully implemented	Method buy_seat was created accordingly. Using this method will allow to buy a plane ticket
4	Fully implemented	Method cancel_seat was created accordingly. Using this method will allow to cancel a already bought ticket.
5	Fully implemented	Method find_first_available was created to find the first unbought steat.
6	Fully implemented	Method show_seating_plan was created to display bought and unbought tickets.

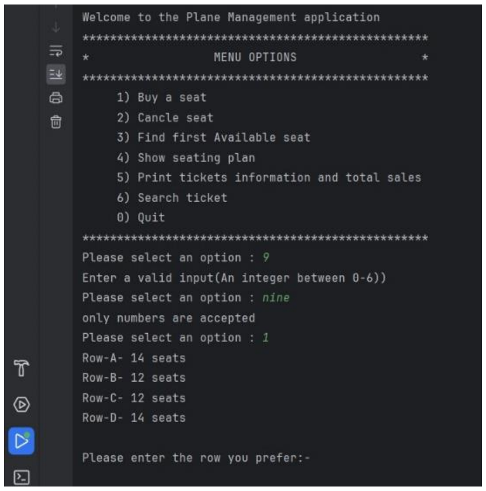
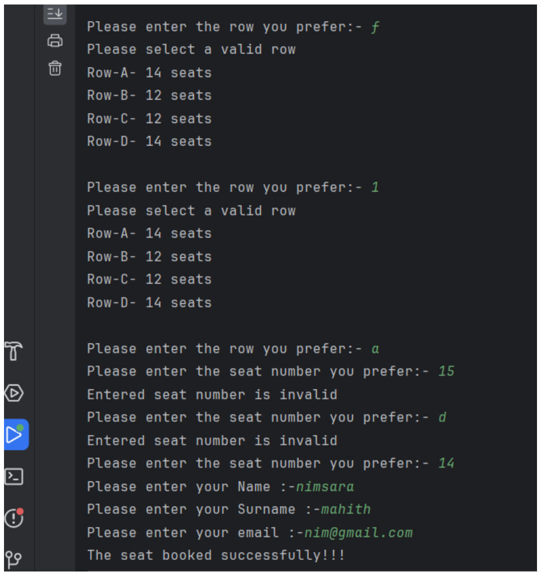


```
Welcome to the Plane Management application
**************************************************
*                  MENU OPTIONS                  *
**************************************************
     1) Buy a seat
     2) Cancle seat
     3) Find first Available seat
     4) Show seating plan
     5) Print tickets information and total sales
     6) Search ticket
     0) Quit
**************************************************
Please select an option : 4
00000000000000
000000000000
000000000000
00000000000000
```

7	Fully implemented	Created the Person.java class and with the following attributes: name, surname, and email. Add a constructor that takes the 3 variables as input to create an object Person. Added all the getters and setters of the class Person. Added a method that prints the information from Person.
8	Fully implemented	Created the Ticket.java class and with the following attributes: row, seat, price, and Person. Person is an object created using the class Person from Task 7. Added all the getters and setters of the class Ticket. Added method that prints the information of a Ticket (including the information of the Person.
9	Fully implemented	Created the ticket array and extended the buy_seat and cancel_seat methods to take appropriate details and to edit the tickets array.
10	Fully implemented	Created the print_ticket_info method to get all the booked seats and the total amount of sales of already bought tickets.
11	Fully implemented	Created the method search_ticket to search for availability of a specific ticket and if the ticket was already sold, information of the buyer.
12	Fully implemented	A text file was created to store the data of a bought ticket.

2) Test Plan

Part A Testing

Test case /scenario	**Input**	**Expected Output**	**Output**	**Pass/Fail**
Selecting the menue option	0>x>6	Warning message		pass
	String entered	Warning message		pass
	0<=x<=6	Continue to the selected method		pass
buy_seat	Incorrect row name	Warning message		pass
	Integer As the row	Warning message		pass
	Correct row letter	Continu to select the seat		pass
	Out of range integer	Warning message		pass
	String value	Warning message		pass
	Correct integer value	Seat will be booked after getting name and email		pass

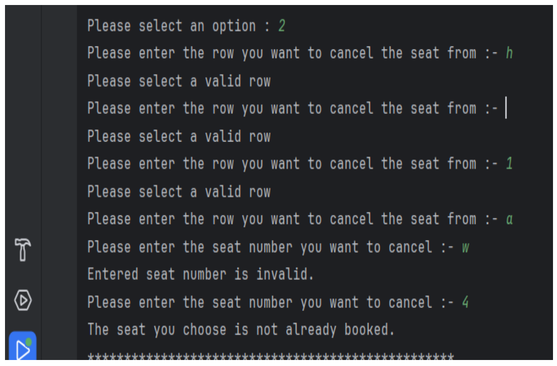
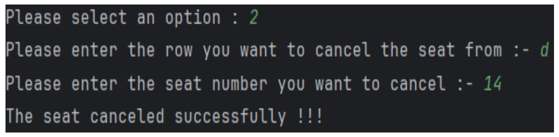
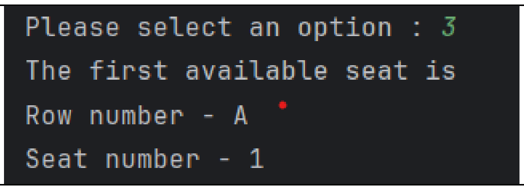
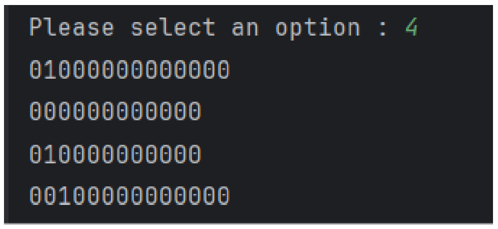
```
Welcome to the Plane Management application
**************************************************
*                  MENU OPTIONS                  *
**************************************************
     1) Buy a seat
     2) Cancle seat
     3) Find first Available seat
     4) Show seating plan
     5) Print tickets information and total sales
     6) Search ticket
     0) Quit
**************************************************
Please select an option : 9
Enter a valid input(An integer between 0-6))
Please select an option : nine
only numbers are accepted
Please select an option : 1
Row-A- 14 seats
Row-B- 12 seats
Row-C- 12 seats
Row-D- 14 seats

Please enter the row you prefer:-
```

```
Please enter the row you prefer:- f
Please select a valid row
Row-A- 14 seats
Row-B- 12 seats
Row-C- 12 seats
Row-D- 14 seats

Please enter the row you prefer:- 1
Please select a valid row
Row-A- 14 seats
Row-B- 12 seats
Row-C- 12 seats
Row-D- 14 seats

Please enter the row you prefer:- a
Please enter the seat number you prefer:- 15
Entered seat number is invalid
Please enter the seat number you prefer:- d
Entered seat number is invalid
Please enter the seat number you prefer:- 14
Please enter your Name :-nimsara
Please enter your Surname :-mahith
Please enter your email :-nim@gmail.com
The seat booked successfully!!!
```

Cancel_seat	Incorrect row name	Warning message		pass
	Integer as the row name	Warning message		pass
	Correct letter	Continu to select the seat		pass
	Out of range integer for seat number	Warning message		pass
	String value	Warning message		pass
	Correct ranged integer	Continue If Booked ,it will be canceled Or Display not booked		pass
find_first_seat_available		Show the first available seat		pass
show_seating_plan		Show the booked seats with “1”and unboocked seats with”0”		pass

```
Please select an option : 2
Please enter the row you want to cancel the seat from :- h
Please select a valid row
Please enter the row you want to cancel the seat from :- |
Please select a valid row
Please enter the row you want to cancel the seat from :- 1
Please select a valid row
Please enter the row you want to cancel the seat from :- a
Please enter the seat number you want to cancel :- w
Entered seat number is invalid.
Please enter the seat number you want to cancel :- 4
The seat you choose is not already booked.
**************************************************
```

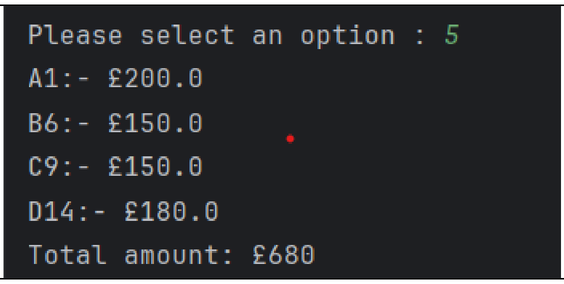
```
Please select an option : 2
Please enter the row you want to cancel the seat from :- d
Please enter the seat number you want to cancel :- 14
The seat canceled successfully !!!
```

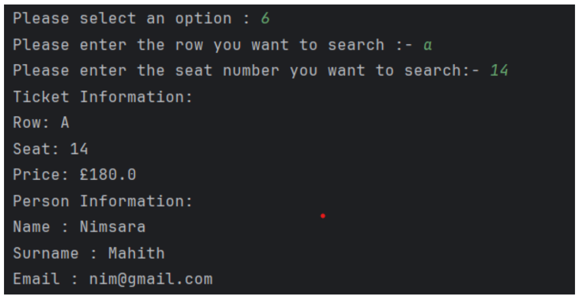
```
Please select an option : 3
The first available seat is
Row number - A
Seat number - 1
```

```
Please select an option : 4
01000000000000
000000000000
010000000000
00100000000000
```

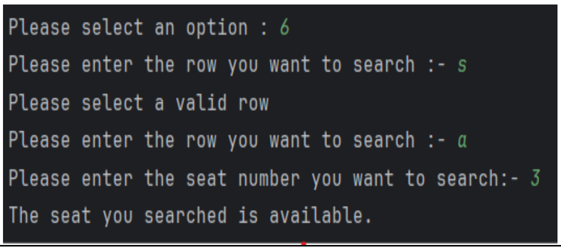
Part B Testing

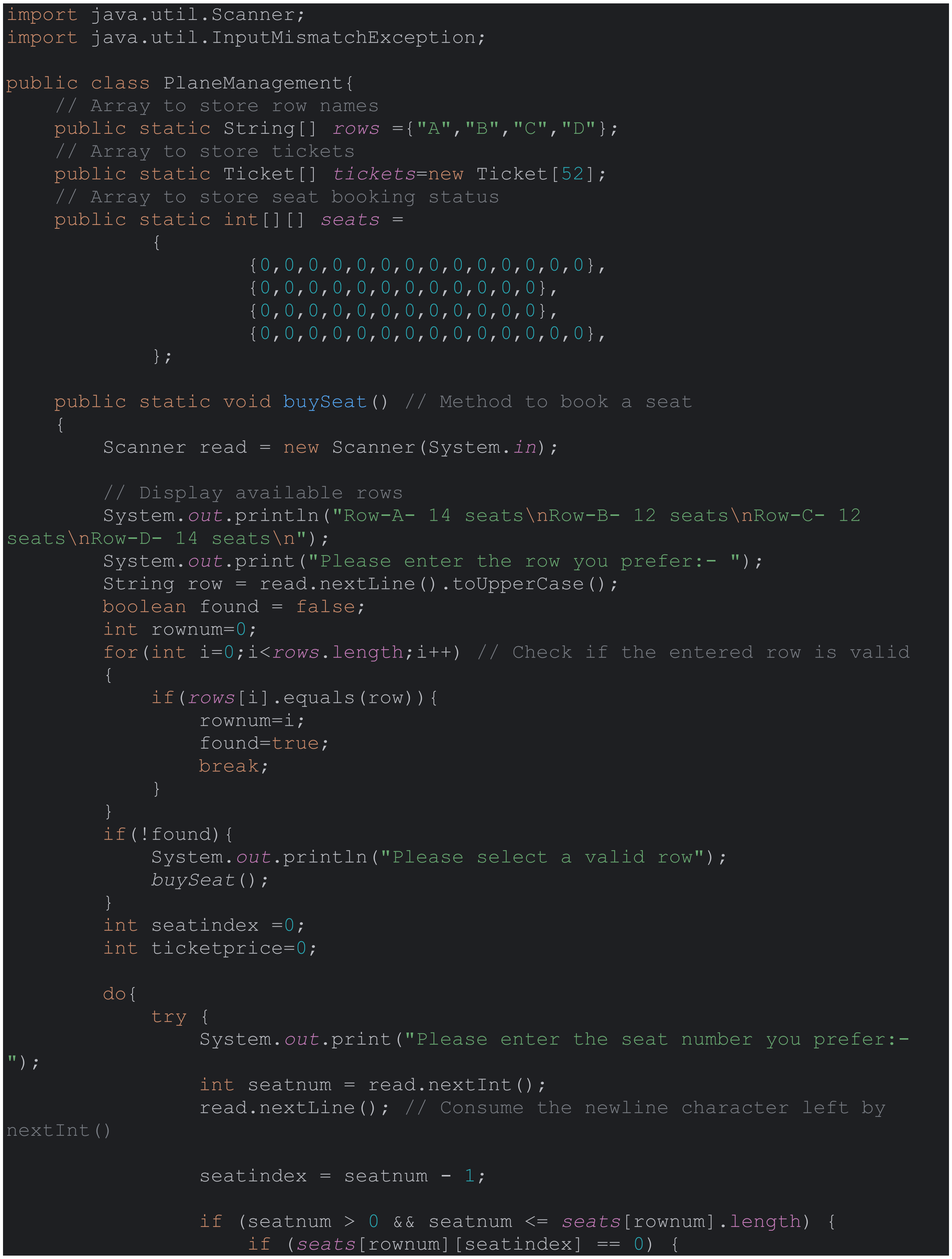
Test case /scenario	Input	Expected Output	Output	Pass/Fail

Print tickets information and total sales	Inputs are taken from the buy_seat	Rows and seat numbers of booked seats and the total sale of them	 `Please select an option : 5`
			`A1:- £200.0`
			`B6:- £150.0`
			`C9:- £150.0`
			`D14:- £180.0`
			`Total amount: £680`
				pass

search ticket	Incorrect row name	Warning message	 `Please select an option : 6`
			`Please enter the row you want to search :- a`
			`Please enter the seat number you want to search:- 14`
			`Ticket Information:`
			`Row: A`
			`Seat: 14`
			`Price: £180.0`
			`Person Information:`
			`Name : Nimsara`
			`Surname : Mahith`
			`Email : nim@gmail.com`
				pass

	Integer as the row name	Warning message		pass
	Correct letter	Continu to select the seat		pass
	Out of range integer for seat number	Warning message		pass
	String value	Warning message		pass

	Correct ranged integer	Continue to, showing details of the booked person if the seat is booked, If the searched ticket isn't booked it will let it know	 `Please select an option : 6`
			`Please enter the row you want to search :- s`
			`Please select a valid row`
			`Please enter the row you want to search :- a`
			`Please enter the seat number you want to search:- 3`
			`The seat you searched is available.`
				pass



```java
import java.util.Scanner;
import java.util.InputMismatchException;

public class PlaneManagement{
    // Array to store row names
    public static String[] rows ={"A","B","C","D"};
    // Array to store tickets
    public static Ticket[] tickets=new Ticket[52];
    // Array to store seat booking status
    public static int[][] seats =
            {
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,0,0},
                    {0,0,0,0,0,0,0,0,0,0,0,0,0,0},
            };

    public static void buySeat() // Method to book a seat
    {
        Scanner read = new Scanner(System.in);

        // Display available rows
        System.out.println("Row-A- 14 seats\nRow-B- 12 seats\nRow-C- 12
seats\nRow-D- 14 seats\n");
        System.out.print("Please enter the row you prefer:- ");
        String row = read.nextLine().toUpperCase();
        boolean found = false;
        int rownum=0;
        for(int i=0;i<rows.length;i++) // Check if the entered row is valid
        {
            if(rows[i].equals(row)){
                rownum=i;
                found=true;
                break;
            }
        }
        if(!found){
            System.out.println("Please select a valid row");
            buySeat();
        }
        int seatindex =0;
        int ticketprice=0;

        do{
            try {
                System.out.print("Please enter the seat number you prefer:-
");

                int seatnum = read.nextInt();
                read.nextLine(); // Consume the newline character left by
nextInt()

                seatindex = seatnum - 1;

                if (seatnum > 0 && seatnum <= seats[rownum].length) {
                    if (seats[rownum][seatindex] == 0) {
                        seats[rownum][seatindex] = 1;

                        // Set ticket price based on seat number
                        if (seatnum<=5){
                            ticketprice = 200;
                        } else if (seatnum<=9) {
                            ticketprice = 150;
                        }else {
                            ticketprice = 180;
                        }

                        System.out.print("Please enter your Name :-");  //
Input passenger details
                        String name = read.nextLine();
                        System.out.print("Please enter your Surname :-");
                        String surname = read.nextLine();
                        System.out.print("Please enter your email :-");
                        String email = read.nextLine();

                        // Create Person object
                        Person person = new Person(name,surname,email);

                        // Create Ticket object
                        Ticket ticket =new
Ticket(row,seatnum,ticketprice,person);
                        for (int i=0;i<tickets.length;i++){ // Add ticket to
tickets array
                            if (tickets[i]==null){
                                tickets[i]=ticket;
                                break;
                            }
                        }
                        System.out.println("The seat booked
successfully!!!");
                        ticket.save(); // Save ticket information to file

                        break;
                    } else {
                        System.out.println("The seat you choose is already
booked");
                    }
                } else {
                    System.out.println("Entered seat number is invalid");
                }
            }catch (InputMismatchException e){
                System.out.println("Entered seat number is invalid");
                read.nextLine();
            }
        }while(seatindex <=0 || seatindex>=seats[rownum].length); // Validate
seat index
        menu(); // Display menu
    }

    // Method to cancel a seat booking
    public static void cancelSeat()
    {
        Scanner read = new Scanner(System.in);
```

```java
        System.out.print("Please enter the row you want to cancel the seat
from :- ");
        String row = read.nextLine().toUpperCase();
        boolean found = false;
        int rownum=0;
        for(int i=0;i<rows.length;i++)   // Check if the entered row is valid
        {
            if(rows[i].equals(row)){ // Validating the row input
                rownum=i;
                found=true;
                break;
            }
        }
        if(!found){
            System.out.println("Please select a valid row");
            cancelSeat();
        }
        int seatindex =0;
        do{
            try {
                System.out.print("Please enter the seat number you want to
cancel :- "); //Validating the seat number input
                int seatnum = read.nextInt();
                seatindex = seatnum - 1;

                if (seatnum > 0 && seatnum <= seats[rownum].length) {
                    if (seats[rownum][seatindex] == 1) {
                        seats[rownum][seatindex] = 0;
                        for (int i=0;i<=tickets.length;i++){ //Removing the
ticket from the array
                            if (tickets[i] != null &&
tickets[i].getRow().equals(row) && tickets[i].getSeatnum() == seatnum) {
                                tickets[i] = null;
                                break;
                            }
                        }
                        System.out.println("The seat canceled successfully
!!!");

                        break;
                    } else {
                        System.out.println("The seat you choose is not
already booked.");
                    }
                } else {
                    System.out.println("Entered seat number is invalid.");
                }
            }catch (InputMismatchException e){
                System.out.println("Entered seat number is invalid.");
                read.nextLine();
            }
        }while(seatindex <=0 || seatindex>=seats[rownum].length);
//Validating the seat index

        menu();
    }
    public static void findFirstAvailableSeat()//Method for finding the first
available seat
    {
        boolean found=false;
        for(int i=0; i<4; i++){ // Looping until finding the seat
            for(int j=0; j<seats[i].length;j++){
                if(seats[i][j]==0){
                    System.out.println("The first available seat is\n"+"Row
number - "+rows[i] +"\nSeat number - "+(j+1)); //Printing the first available
seat
                    found=true;
                    break;
                }
            }
            if(found){
                menu();
            }
        }
        if(!found){
            System.out.println("There's no available seats.");
        }
    }
    public static void showSeatPlan() // Method for showing the seat plan
(available)
    {
        for(int i=0; i<4; i++){ //Looping through the seats array to print
the layout
            for(int j=0; j<seats[i].length;j++){
                System.out.print(seats[i][j]+"");
            }
            System.out.println();
        }
        menu();
    }
    public static void printTicketInfo() { // The method for printing the
already bought tickets and the total sales
        int totalAmount = 0;
        int ticketCount = 0;

        for (Ticket ticket : tickets) { // Getting the seat count (purchased)
            if (ticket != null) {
                ticketCount++;
            }
        }

        for (int i = 0; i < ticketCount; i++) {   // Print details of each
booked ticket
            Ticket currentTicket = tickets[i];
            if (currentTicket != null) {
                String row = currentTicket.getRow();
                int seatnum = currentTicket.getSeatnum();
                double ticketPrice = currentTicket.getPrice();
                System.out.println(row + seatnum + ":- £" + ticketPrice); //
Print ticket details
                totalAmount += (int) ticketPrice; // Update total amount of
sales
            }
        }
        // Print total amount of ticket sales
        System.out.println("Total amount: £" + totalAmount);
        menu();
    }

    public static void searchTicket() //The method for searching for 
available seats
    {
        Scanner read = new Scanner(System.in);

        int ticketCount=0;
        for (Ticket ticket : tickets) {     // Count the number of booked 
tickets
            if (ticket != null) {
                ticketCount+=1;
            }
        }
        // Prompt user to enter row to search
        System.out.print("Please enter the row you want to search :- ");
        String row = read.nextLine().toUpperCase();
        boolean found = false;
        int rownum=0;

        for(int i=0;i<rows.length;i++) // Check if the entered row is valid
        {
            if(rows[i].equals(row)){
                rownum=i;
                found=true;
                break;
            }
        }
        if(!found){  // If entered row is not valid, prompt user to enter 
again
            System.out.println("Please select a valid row");
            searchTicket();
        }
        int seatindex=0;
        // Prompt user to enter seat number to search
        do{
            try {
                System.out.print("Please enter the seat number you want to 
search:- ");
                int seatnum = read.nextInt();
                seatindex = seatnum - 1;

                // Check if entered seat number is within valid range
                if (seatnum >= 0 && seatnum <= seats[rownum].length) {
                    if (seats[rownum][seatindex] == 1) {
                        for (int i=0;i<=ticketCount;i++) {  // If seat is 
booked, find and print ticket information
                            if (tickets[i].getRow().equals(row)&&
tickets[i].getSeatnum()==seatnum) {
                                tickets[i].printTicketInfo();
                                menu();
                            }
                        }
                    }else { //Showing the searched seat is available
                        System.out.println("The seat you searched is
available.");
                        break;
                    }
                } else {
                    System.out.println("Entered seat number is invalid.");
                }
            }catch (InputMismatchException e){
                System.out.println("Entered seat number is invalid.");
                read.nextLine(); // Clear input buffer
            }
        }while(seatindex <=0 || seatindex>=seats[rownum].length);
        menu();
    }

    public static void sc(){ // Method to select options
        Scanner read = new Scanner(System.in);

        boolean correct = false;
        // Loop until correct option is selected
        while (!correct){
            try{  // Prompt user to select an option
                System.out.print("Please select an option : ");
                int option= read.nextInt();
                switch (option) {
                    case 1:
                        buySeat();
                        break;
                    case 2:
                        cancelSeat();
                        break;
                    case 3:
                        findFirstAvailableSeat();
                        break;
                    case 4:
                        showSeatPlan();
                        break;
                    case 5:
                        printTicketInfo();
                        break;
                    case 6:
                        searchTicket();
                        break;
                    case 0:
                        System.out.println("Exited the the Plane Management
application. ");
                        break;
                    default:
                        System.out.println("Enter a valid input(An integer
between 0-6))");
                        sc(); // Recursive call to allow user to enter valid
input
                }
                correct=true;  // Set flag to exit loop
            }catch (InputMismatchException e){
                System.out.println("only numbers are accepted");
                sc(); // Recursive call to allow user to enter valid input
                break; // Exit loop
            }
        }
    }
    public static void menu(){ // Method to print the menue

        for (int r1 = 1; r1 <= 50; r1++) {
            System.out.print("*");
        }
        System.out.println();
        System.out.println("*                  MENU OPTIONS
*");

        for (int r1 = 1; r1 <= 50; r1++) {
            System.out.print("*");
        }
        System.out.println();
        System.out.println("     1) Buy a seat");
        System.out.println("     2) Cancle seat");
        System.out.println("     3) Find first Available seat");
        System.out.println("     4) Show seating plan");
        System.out.println("     5) Print tickets information and total
sales");
        System.out.println("     6) Search ticket");
        System.out.println("     0) Quit");

        for (int r1 = 1; r1 <= 50; r1++) {
            System.out.print("*");
        }
        System.out.println();
        sc();
    }
    public static void main(String[] args) { // main method
        System.out.println("Welcome to the Plane Management application");
        menu();
    }
}
```

```
public class Person { //Person class

    //convirting the class details to a string
    public String toString() {
        return " Name :- " + name + "\n Surname :- " + surname + "\n Email :-
" + email+"\n";
    }
    private String name;
    private String surname;
    private String email;

    // The constructor.
    public Person(String name, String surname, String email){
        this.name=name;
        this.surname=surname;
        this.email=email;
    }
    // Getters and setters
    public String getName(){
        return name;
    }
    public String getSurname(){
        return surname;
    }
    public String getEmail(){
        return email;
    }
    public void setName(String name){
        this.name=name;
    }
    public void setSurname(String surname){
        this.surname=surname;
    }
    public void setEmail(String email){
        this.email=email;
    }

    //Method to print the person details.
    public void printDetails(){
        System.out.println("Name : "+name);
        System.out.println("Surname : "+surname);
        System.out.println("Email : "+email);
    }

}
```

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
public class Ticket { // Ticket class
```

```java
    private String row;
    private int seatnum;
    private double price;
    private Person person;

    //Conviriting to a string
    public String toString() {
        return "Ticket details\n Row :- " + row + "\n Seat :- " + seatnum +
"\n Price :- £" + price + "\n" + person.toString();
        }

        //The constructor
    public Ticket(String row, int seat, double price, Person person) {
        this.row = row;
        this.seatnum = seat;
        this.price = price;
        this.person = person;
    }
    //Getters and setters
    public String getRow() {
        return row;
    }
    public void setRow(String row) {
        this.row = row;
    }
    public int getSeatnum() {
        return seatnum;
    }
    public void setSeatnum(int seatnum) {
        this.seatnum = seatnum;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public Person getPerson() {
        return person;
    }
    public void setPerson(Person person) {
        this.person = person;
    }

    //Method to print ticket details.
    public void printTicketInfo() {
        System.out.println("Ticket Information:");
        System.out.println("Row: " + row);
        System.out.println("Seat: " + seatnum);
        System.out.println("Price: £" + price);
        System.out.println("Person Information:");
        person.printDetails();
    }
    // Method to save the text file
    public void save() {
        try {
            File file = new File(row + seatnum + ".txt");
            FileWriter writer = new FileWriter(file);
            writer.write("Row: " + row + "\n");
            writer.write("Seat: " + seatnum + "\n");
            writer.write("Price: £ " + price + "\n");
            writer.write("Name: " + person.getName() + "\n");
            writer.write("Surname: " + person.getSurname() + "\n");
            writer.write("Email: " + person.getEmail() + "\n");
            writer.close();
        } catch (IOException e) {
            System.out.println("Error saving ticket information: " +
e.getMessage());
        }
    }

}
```