Informatics Institute of Technology

School of Computing

Software Development II Coursework Report

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

Date of submission : 23rd March 2023

Student ID : IIT No- 20230039 UOW No-2051518

Student First Name : Nihinsa Lineli

Student Surname : Wijesinghe

Tutorial group : Group 1

Day, time, and tutor/s: Monday, 1.30p.m.-3.30p.m., Mr.Pumudu Fernando

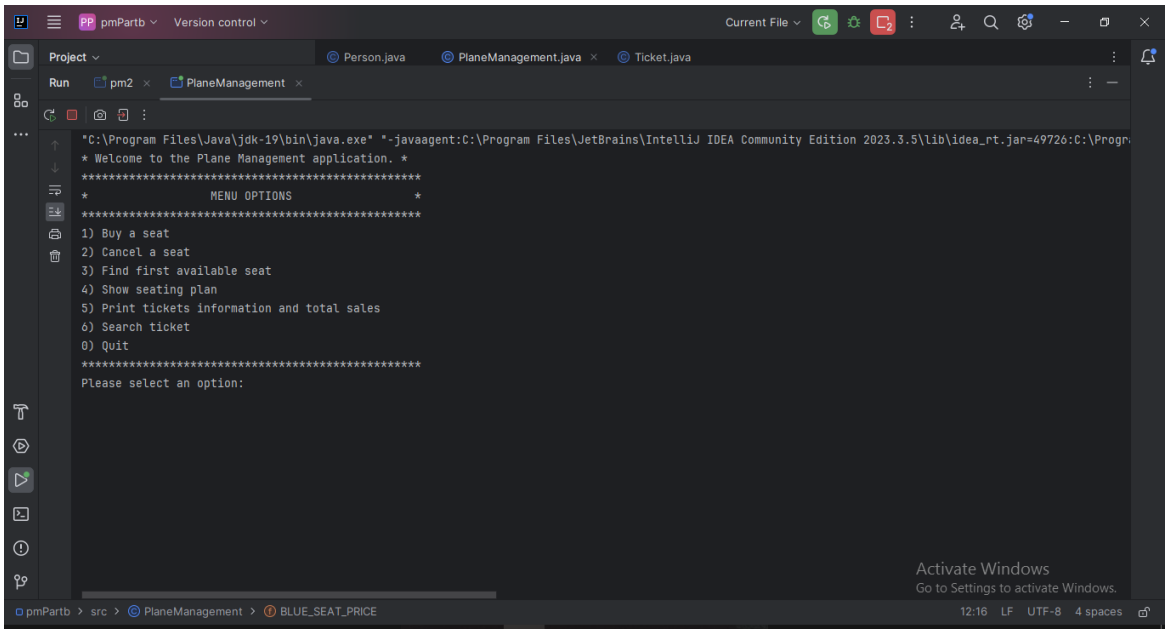
"I confirm that I understand what plagiarism / collusion / contract cheating is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged."

Name : Nihinsa Lineli Wijesinghe

Student ID : IIT No- 20230039 UOW No-2051518

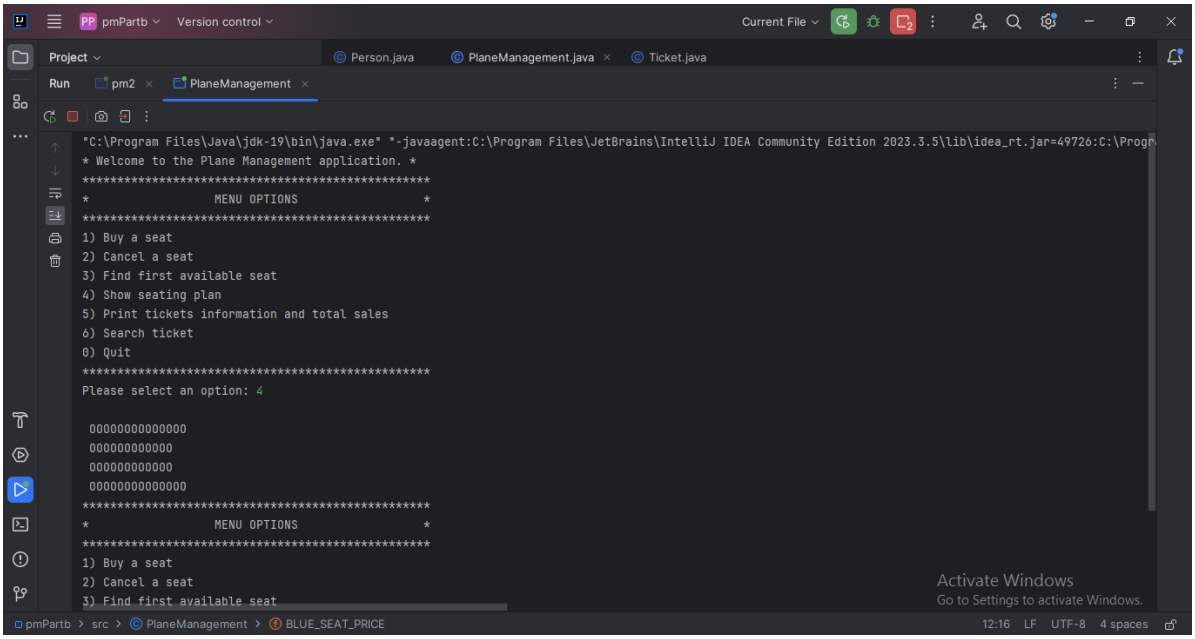
Self-assessment form and test plan

1) Self-assessment form

Task	Self-assessment (select one)	Comments
1	☒Fully implemented ☐Partially implemented ☐Not attempted	
2	☒Fully implemented ☐Partially implemented ☐Not attempted	
Insert here a screenshot of your welcome message and menu:  		
3	☒Fully implemented ☐Partially implemented ☐Not attempted	
4	☒Fully implemented ☐Partially implemented ☐Not attempted	
5	☒Fully implemented ☐Partially implemented ☐Not attempted	
6	☒Fully implemented	

	☐Partially implemented ☐Not attempted	

Insert here a screenshot of the seating plan:



7	☒Fully implemented ☐Partially implemented ☐Not attempted	
8	☒Fully implemented ☐Partially implemented ☐Not attempted	
9	☒Fully implemented ☐Partially implemented ☐Not attempted	
10	☒Fully implemented ☐Partially implemented ☐Not attempted	
11	☒Fully implemented ☐Partially implemented ☐Not attempted	
12	☒Fully implemented ☐Partially implemented ☐Not attempted	

2) Test Plan

Complete the test plan describing which testing you have performed on your program.

Add as many rows as you need.

Part A Testing

Test case / scenario	Input	Expected Output	Output	Pass/Fail
1.Menu	Start the program	Print user menu	Print the entire user menu	☒Pass ☐Fail
2.Buy a seat	Option:1 Row: A Seat: 5	Seat successfully reserved. Total price: £200	Seat successfully reserved. Total price: £200	☒Pass ☐Fail
	Option:1 Row: Y	Invalid row. Please try again.	Invalid row. Please try again.	☒Pass ☐Fail
	Option:1 Row: A Seat: Y	Invalid input. Please enter a valid integer.	Invalid input. Please enter a valid integer.	☒Pass ☐Fail
	Option:1 Row: A Seat: 33	Invalid seat number. Please try again.	Invalid seat number. Please try again.	☒Pass ☐Fail
	Option:1 Row: A Seat: 5	Seat is already reserved. Please select a different seat.	Seat is already reserved. Please select a different seat.	☒Pass ☐Fail
3.Cancel a seat	Option:2 Row: A Seat: 5	Seat successfully cancelled.	Seat successfully cancelled.	☒Pass ☐Fail
	Option:2 Row: A Seat: 5	Seat is already available. No need to cancel.	Seat is already available. No need to cancel.	☒Pass ☐Fail
	Option:2 Row: Y	Invalid row. Please try again.	Invalid row. Please try again.	☒Pass ☐Fail
	Option:2 Row: A Seat: Y	Invalid input. Please enter a valid integer.	Invalid input. Please enter a valid integer.	☒Pass ☐Fail
	Option:2 Row: A Seat: 33	Invalid seat number. Please try again.	Invalid seat number. Please try again.	☒Pass ☐Fail
4.Find the first available seat	Option:3	First available seat: A1	First available seat: A1	☒Pass ☐Fail
	Buy Row: A, Seat: 1 then select option:3	First available seat: A2	First available seat: A2	☒Pass ☐Fail
	Cancel Row: A,	First available seat: A1	First available seat: A1	☒Pass ☐Fail

	Seat: 1 then select option:3			
	Book all seats then select option 3:	No available seats found.	No available seats found.	☒Pass ☐Fail
5.Show Seating Plan	Option:4	OOOOOOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOOOO	OOOOOOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOOOO	☒Pass ☐Fail
	Book Row: A, Seat: 5 then select option:4	OOOOXOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOOOO	OOOOXOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOO OOOOOOOOOOOOOO	☒Pass ☐Fail
	Manually book all seats then select option:4	XXXXXXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXXXX	XXXXXXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXX XXXXXXXXXXXXXX	☒Pass ☐Fail
6.Quit	Option-0	Thank you for using our plane management application. Safe travels!	Thank you for using our plane management application. Safe travels!	☒Pass ☐Fail

Part B testing

Test case / scenario	**Input**	**Expected Output**	**Output**	**Pass/Fail**
1.Buy a seat	Option:1 Row: A Seat: 5	Enter passenger name: Enter passenger surname: Enter passenger email:	Enter passenger name: Enter passenger surname: Enter passenger email:	☒Pass ☐Fail

		Seat successfully reserved. Total price: £200	Seat successfully reserved. Total price: £200	
2.Print ticket info	Buy the tickets of Row: A Seat: 5 Row: B Seat: 3 Row: C Seat: 12 Then select Option-5	Ticket Information: Total number of tickets purchased: 3 Ticket 1- Row: A Seat: 5 Price: $200.0 Passenger Information- Passenger name: Passenger surname: Passenger email address: Ticket 2- Row: B Seat: 3 Price: $200.0 Passenger Information- Passenger name: Passenger surname: Passenger email address: Ticket 3- Row: C Seat: 12 Price: $180.0 Passenger Information- Passenger name: Passenger surname: Passenger email address: Total amount for tickets sold: £580.0	Ticket Information: Total number of tickets purchased: 3 Ticket 1- Row: A Seat: 5 Price: $200.0 Passenger Information- Passenger name: w Passenger surname: w Passenger email address: w Ticket 2- Row: B Seat: 3 Price: $200.0 Passenger Information- Passenger name: Passenger surname: Passenger email address: Ticket 3- Row: C Seat: 12 Price: $180.0 Passenger Information- Passenger name: Passenger surname: Passenger email address: Total amount for tickets sold: £580.0	☒Pass ☐Fail
3.Search ticket	Option:6 Row: A	This seat is available.	This seat is available.	☒Pass ☐Fail

	Seat: 1			
	Buy Row:A Seat:2 then select Option:1 Row: A Seat: 1	Sorry,selected seat has already been purchased. Ticket Information of the purchased seat- Row: A Seat : 1 Price: $200.0 Passenger Information- Passenger name: Passenger surname: Passenger email address:	Sorry,selected seat has already been purchased. Ticket Information of the purchased seat- Row: A Seat : 1 Price: $200.0 Passenger Information- Passenger name: Passenger surname: Passenger email address:	☒Pass ☐Fail
	Option:6 Row: Y	Invalid row. Please try again.	Invalid row. Please try again.	☒Pass ☐Fail
	Option:6 Row: A Seat: Y	Invalid input. Please enter a valid integer.	Invalid input. Please enter a valid integer.	☒Pass ☐Fail
	Option:6 Row: A Seat: 33	Invalid seat number. Please try again.	Invalid seat number. Please try again.	☒Pass ☐Fail
4.Save ticket info to a file	Option:1 Row: A Seat: 5	Seat successfully reserved. Total price: £200 Ticket information saved to file: A5.txt	Seat successfully reserved. Total price: £200 Ticket information saved to file: A5.txt	☒Pass ☐Fail
5.Delete the ticket file when a seat is cancelled	Option:2 Row: A Seat: 5	Seat successfully cancelled. Ticket file deleted.	Seat successfully cancelled. Ticket file deleted.	☒Pass ☐Fail

Are there any specific parts of the coursework which you would like to get feedback?

You will need to demonstrate your understanding of the submitted code. Your tutor will arrange a coursework demonstration. During the coursework demonstration, your tutor will ask you to execute your program and questions on your code.

Failure to attend the demonstration will result in <u>0 for the coursework.</u>

3) Code :

Plane management class:

```
import java.util.*;


public class w2051518_PlaneManagement {

    //constants for the program

    private static final int ROWS = 4;

    private static final int[] SEATS_PER_ROW = {14, 12, 12, 14};


    private static final int AVAILABLE = 0;

    private static final int SOLD = 1;


    private static final int YELLOW_SEAT_PRICE = 200;

    private static final int BLUE_SEAT_PRICE = 150;

    private static final int GREEN_SEAT_PRICE = 180;


    //Array to store the status of the seats

    private static int[][] seats = new int[ROWS][];

    // Array to store sold tickets

    private static Ticket[] soldTickets = new Ticket[ROWS * SEATS_PER_ROW.length];
```

```
private static int ticketCount = 0;

private static double totalSales=0;

private static Scanner pm = new Scanner(System.in);


// Method to print user menu

public static void printUserMenu() {

    for (int i = 0; i < 50; i++) {

        System.out.print("*");

    }

    System.out.println();

    System.out.println("*                 MENU OPTIONS                 *");

    for (int i = 0; i < 50; i++) {

        System.out.print("*");

    }

    System.out.println();

    System.out.println("1) Buy a seat");

    System.out.println("2) Cancel a 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 i = 0; i < 50; i++) {
        System.out.print("*");
    }
    System.out.println();
}
// Method to store the seat layout
public static void seat_status() {
    for (int i = 0; i < ROWS; i++) {
        seats[i] = new int[SEATS_PER_ROW[i]];
        for (int j = 0; j < SEATS_PER_ROW[i]; j++) {
            seats[i][j] = AVAILABLE;
        }
    }
}
// Method to buy a seat
public static void buy_seat() {
    int row = input_row();
    if (row == -1) {
        return;
    }int seatNum = input_seatNum(row);
    if (seatNum == -1) {
        return;
```

```
        }
        if (!seat_availability(row, seatNum)) {
            System.out.println("Seat is already reserved. Please select a
different seat.");
            return;
        }
        //Ask for Person information
        System.out.print("Enter passenger name: ");
        String name = pm.next();
        System.out.print("Enter passenger surname: ");
        String surname = pm.next();
        System.out.print("Enter passenger email: ");
        String email = pm.next();




        Person person = new Person(name, surname, email);
        int price = calculate_price(row, seatNum);


        Ticket ticket = new Ticket((char) ('A' + row), seatNum + 1, price,
person);
        // Add the ticket to the  soldTickets array
        soldTickets[ticketCount++] = ticket;
        // Update seat status
```

```
        seats[row][seatNum] = SOLD;

        totalSales += price;


        System.out.println("Seat successfully reserved. Total price: £" +
price);

        ticket.save();

    }

    //Get a valid row letter

    public static int input_row() {

        while (true) {

            System.out.println("Please select the row (A, B, C, D): ");

            String input = pm.next().toUpperCase(); // Convert to uppercase

            if (input.length() != 1 || input.charAt(0) < 'A' || input.charAt(0)
> 'D') {

                System.out.println("Invalid row. Please try again.");

            } else {

                return input.charAt(0) - 'A'; // Convert row letter to index

            }

        }

    }

    //Get a valid seat number

    public static int input_seatNum(int row) {

        while (true) {
```

```
            System.out.print("Enter seat number: ");
            if (!pm.hasNextInt()) {
                System.out.println("Invalid input. Please enter a valid integer.");
                pm.next(); // Discard the invalid input
            } else {
                int seatNum = pm.nextInt() - 1;
                if (seatNum < 0 || seatNum >= SEATS_PER_ROW[row]) {
                    System.out.println("Invalid seat number. Please try again.");
                } else {
                    return seatNum;
                }
            }
        }
    }
    // Method to check seat availability
    public static boolean seat_availability(int row, int seatNum) {
        return seats[row][seatNum] == AVAILABLE;
    }
    // Method to cancel a seat
    public static void cancel_seat() {
        System.out.println("Please select the seat to cancel:");
```

```
        int row = input_row();

        if (row == -1) {

            return;

        }int seatNum = input_seatNum(row);

        if (seatNum == -1) {

            return;

        }

        if (seat_availability(row, seatNum)) {

            System.out.println("Seat  is  already  available.  No  need  to
cancel.");

            return;

        }

        seats[row][seatNum]  =  AVAILABLE;//  Resetting  the  seat  status  to
available after canceling a reservation

        System.out.println("Seat successfully cancelled.");



        for (int i = 0; i < ticketCount; i++) {

            Ticket ticket = soldTickets[i];

            if (ticket.getRow() == ('A' + row) && ticket.getSeat() == seatNum
+ 1) {

                for (int j = i; j < ticketCount - 1; j++) {

                    soldTickets[j] = soldTickets[j + 1];

                }

                soldTickets[ticketCount - 1] = null; // Remove the last element
```

```
            ticketCount--;
            ticket.deleteFile();
            break;
        }
    }
}
//Method to calculate the price of a seat according to the respective colour
public static int calculate_price(int row, int seatNum) {
    if (seatNum >= 0 && seatNum < 5) {
        return YELLOW_SEAT_PRICE;
    } else if (seatNum >= 5 && seatNum < 9) {
        return BLUE_SEAT_PRICE;
    } else {
        return GREEN_SEAT_PRICE;
    }
}
//Method to find the first available seat
public static void find_first_available() {
    for (int row = 0; row < ROWS; row++) {
        for (int seatNum = 0; seatNum < SEATS_PER_ROW[row]; seatNum++) {
            if (seat_availability(row, seatNum)) {
                char rowLetter = (char) ('A' + row);
```

```
                    System.out.println("First available seat: " + rowLetter +
(seatNum + 1));
                    return;
                }
            }
        }System.out.println("No available seats found.");
    }
    //Method to show the seating plan
    public static void show_seating_plan() {
        System.out.println();
        for (int row = 0; row < ROWS; row++) {
            System.out.print(" ");
            for (int seatNum = 0; seatNum < SEATS_PER_ROW[row]; seatNum++) {
                if (seat_availability(row, seatNum)) {
                    System.out.print('O');
                } else {
                    System.out.print('X');
                }
            }System.out.println();
        }
    }
    //Method to print the information of the tickets
    public static void print_tickets_info() {
```

```
        System.out.println("Ticket Information:");

        System.out.println("Total number of tickets purchased: " +
ticketCount);

        double totalPrice = 0;

        for (int i = 0; i < ticketCount; i++) {

            Ticket ticket = soldTickets[i];

            System.out.println("Ticket " + (i + 1) + "-");

            System.out.println("Row: " + ticket.getRow());

            System.out.println("Seat: " + ticket.getSeat());

            System.out.println("Price: $" + ticket.getPrice());

            ticket.getPerson().printInfo();

            System.out.println();

        }System.out.println("Total amount for tickets sold: £" + totalSales);

        for (int i = 0; i < ticketCount; i++) {

            Ticket ticket = soldTickets[i];

        }

    }

    //Method to search for a ticket

    public static void search_ticket(){

        System.out.println("Enter seat to search: ");

        int row = input_row();

        if (row == -1) {

            return;
```

```
        }int seatNum = input_seatNum(row);

        if (seatNum == -1) {

            return;

        }

        if (!seat_availability(row, seatNum)) {

            for (Ticket ticket : soldTickets) {

                if (ticket != null && ticket.getRow() == ('A' + row) &&
ticket.getSeat() == seatNum + 1) {

                    System.out.println("Sorry,selected seat has already been
purchased.");

                    System.out.println("Ticket Information of the purchased
seat-");

                    ticket.printInfo();

                }return;

            }

        }System.out.println("This seat is available.");

    }

    //Main method

    public static void main(String[] args) {

        seat_status();

        System.out.println("* Welcome to the Plane Management application. *");

        while (true) {

            printUserMenu();

            System.out.print("Please select an option: ");
```

```
            try {
                int option = pm.nextInt();
                switch (option) {
                    case 0:
                        System.out.println("Thank you for using our plane
management application. Safe travels!");
                        return;
                    case 1:
                        buy_seat();
                        break;
                    case 2:
                        cancel_seat();
                        break;
                    case 3:
                        find_first_available();
                        break;
                    case 4:
                        show_seating_plan();
                        break;
                    case 5:
                        print_tickets_info();
                        break;
                    case 6:
```

```
                        search_ticket();
                        break;
                    default:
                        System.out.println("Invalid   option.   Please   try
again.");
                }
            } catch (InputMismatchException e) {
                System.out.println("Invalid option. Please try again.");
            }pm.nextLine(); // Clear the input buffer
        }
    }
}
```

Ticket class:

```
import java.io.IOException;
import java.io.FileWriter;
import java.io.File;
public class Ticket {
    private char row;
    private int seat;
    private double price;
    private Person person;

    public Ticket(char row, int seat, double price, Person person){
```

```
    this.row=row;

    this.seat=seat;

    this.price=price;

    this.person=person;

}


public char getRow(){

    return row;

}

public void setRow(char row) {

    this.row = row;

}


public int getSeat(){

    return seat;

}

public void setSeat(int seat){

    this.seat=seat;

}


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 information
public void printInfo() {
    System.out.println("Row: " + row);
    System.out.println("Seat : " + seat);
    System.out.println("Price: $" + price);


    person.printInfo();
}
//Method to save the ticket information to a file
public void save() {
```

```
        String filename = row + String.valueOf(seat) + ".txt";

        try {

            FileWriter saveTicket = new FileWriter(filename);

            saveTicket.write("Ticket Information:\n");

            saveTicket.write("Row: " + row + "\n");

            saveTicket.write("Seat : " + seat + "\n");

            saveTicket.write("Price: $" + price + "\n");

            saveTicket.write("Passenger Information-\n");

            saveTicket.write("Passenger name: " + person.getName() + "\n");

            saveTicket.write("Passenger  surname:  "  +  person.getSurname()  +
"\n");

            saveTicket.write("Passenger email address: " + person.getEmail() +
"\n");

            saveTicket.close();

            System.out.println("Ticket  information  saved  to  file:  "  +
filename);

        } catch (IOException e) {

            System.out.println("An  error  occurred  while  saving  the  ticket
information to file.");

            e.printStackTrace();

        }

    }

    //Method to delete the ticket file
```

```
    public void deleteFile() {

        String filename = row + String.valueOf(seat) + ".txt";

        File file = new File(filename);

        if (file.exists()) {

            if (file.delete()){

            System.out.println("Ticket file deleted." );

            } else {

                System.out.println("Failed to delete the ticket file." );

            }

        }else{

            System.out.println("Ticket file does not exist." );

        }

    }

}
```

Person class:

```
public class Person {

    private String name;

    private String surname;

    private String email;
```

```
public Person(String name, String surname, String email) {

    this.name = name;

    this.surname = surname;

    this.email = email;

}


public String getName() {

    return name;

}

public void setName(String name) {

    this.name = name;

}


public String getSurname() {

    return surname;

}

public void setSurname(String surname) {

    this.surname = surname;

}


public String getEmail() {
```

```
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }


    // Method to print person information
    public void printInfo() {
        System.out.println("Passenger Information-");
        System.out.println("Passenger name: " + name);
        System.out.println("Passenger surname: " + surname);
        System.out.println("Passenger email address: " + email);
    }
}
```

<<END>>