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

public class w2052215_PlaneManagement {

  private Ticket ticket;
  private 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}
  };
  private static Ticket[][] tickets = new Ticket[4][14]; // 2D array to store tickets

  public static void main(String[] args) {
    System.out.println("Welcome to the Plane Management application");
    Scanner scanner = new Scanner(System.in);

    int option = 1; //1 is set to options initially to ensure that the loop runs at least onc

    do {
      System.out.println("**************************************************");
      System.out.println("*                  MENU OPTIONS                  *");
      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 ticket information and total sales");
      System.out.println("     6) Search tickets");
      System.out.println("     0) Quit");
      System.out.println("**************************************************");
      System.out.println("Please select an option");

      try {
        option = scanner.nextInt();

        switch (option) {
          case 1:
            buySeat(scanner);
            break;
          case 2:
            cancelSeat(scanner);
            break;
          case 3:
            findAvailableSeat();
            break;
          case 4:
            displaySeats();
            break;
          case 5:
            printTicketInfo();
            break;
          case 6:
            searchTickets(scanner);
            break;
          case 0:
```

```
                    System.out.println("Exiting program");
                    break;
                default:
                    System.out.println("Invalid option. Please try again.");
            }
        } catch (InputMismatchException ex) {
            System.out.println("Invalid input. Please enter a valid number.");
            scanner.nextLine(); // Clear the input buffer
        }
    } while (option != 0);
}

public static void buySeat(Scanner scanner) {
    // Ask the user for input
    System.out.println("Enter the row letter (A-D): ");
    char rowLetter = scanner.next().toUpperCase().charAt(0); // Convert input to uppercase
    System.out.println("Enter the seat number (1-14): ");
    int seatNumber = scanner.nextInt();


    // Validate row and seat number
    if(rowLetter < 'A' | rowLetter > 'D'){System.out.println("Invalid row or seat number.");
        return;
    }


    else if(rowLetter =='A' || rowLetter =='D'){
        if(seatNumber < 1 || seatNumber >14){
            System.out.println("Invalid row or seat number.");
            return;
        }
    }else if(rowLetter == 'B' || rowLetter == 'C'){
        if(seatNumber < 1 || seatNumber > 12){
            System.out.println("Invalid row or seat number.");
            return;
        }
    }

    // Convert row letter to array index
    int row = rowLetter - 'A';

    // Check if the seat is available
    if (seats[row][seatNumber - 1] == 1) {
        System.out.println("Seat is already occupied. Please choose another seat.");
    } else {
        // Mark the seat as sold
        seats[row][seatNumber - 1] = 1;

        // Ask for person information
        System.out.println("Enter person's name:");
        String name = scanner.next();
        System.out.println("Enter person's surname:");
        String surname = scanner.next();
        System.out.println("Enter person's email:");
        String email = scanner.next();

        // Create a Person object
```

```
            person person = new person(name, surname, email);

            // Define price based on seat location
            double price = calculatePrice(row, seatNumber);

            // Create a Ticket object
            Ticket ticket = new Ticket(row, seatNumber, price, person);

            // Add the ticket to the tickets array
            tickets[row][seatNumber - 1] = ticket;

            ticket.save();

            System.out.println("Seat " + rowLetter + seatNumber + " has been successfully sold.");
        }
    }

    private static double calculatePrice(int row, int seatNumber) {
        double price = 0;
        if(seatNumber <= 5) {
            price = 200;
        }
           else if(seatNumber >= 6 && seatNumber <= 9){
                price = 150;
            }
                else{
                    price = 180;
        }
        return price;
    }


    private static void cancelSeat(Scanner scanner) {

        // Implement cancelSeat method to remove the ticket from the array of tickets
        System.out.println("Enter row letter (A-D): ");
        char rowLetter = scanner.next().toUpperCase().charAt(0);
        int row = rowLetter - 'A';
        if (row < 0 || row >= seats.length) {
            System.out.println("Invalid row letter. Please try again.");
            return;
        }

        System.out.println("Enter seat number: ");
        int seatNumber = scanner.nextInt();

        // Validate row and seat number
        if (seatNumber < 1 || seatNumber > seats[row].length) {
            System.out.println("Invalid seat number. Please try again.");
            return;
        }

        // Check if the seat is available
        if (seats[row][seatNumber - 1] == 0) {
            System.out.println("Seat is already available. Please choose another seat to cancel.");
        } else {
            seats[row][seatNumber - 1] = 0; // Mark the seat as available
            tickets[row][seatNumber - 1] = null; // Remove the ticket from the tickets array
            System.out.println("Seat " + rowLetter + seatNumber + " canceled successfully.");
        }
    }

    private static void findAvailableSeat() {
        boolean seatFound = false;

        for (int i = 0; i < seats.length; i++) {
            for (int j = 0; j < seats[i].length; j++) {
                if (seats[i][j] == 0) {
                    char rowLetter = (char) ('A' + i);
                    System.out.println("First available seat found: " + rowLetter + (j + 1));
                    seatFound = true;
                    break;
                }
            }
            if (seatFound) {
                break;
            }
        }

        if (!seatFound) {
            System.out.println("No available seats found.");
        }
    }

    private static void displaySeats() {
        System.out.println("Seating Plan:");

        for (int i = 0; i < seats.length; i++) {
            if (i == 1 || i == 2) {
                System.out.print(" "); // Add a space at the start of row B and C
            }

            for (int j = 0; j < seats[i].length; j++) {
                if (seats[i][j] == 0) {
                    System.out.print("O");
                } else {
                    System.out.print("X");
                }
            }
            System.out.println(); // Move to the next row after printing seats in the current row
        }
    }

    private static void printTicketInfo() {
        double totalPrice = 0;

        // Iterate through the tickets array to print ticket information and calculate total price
        for (int i = 0; i < tickets.length; i++) {
            for (int j = 0; j < tickets[i].length; j++) {
                if (tickets[i][j] != null) { // Check if the seat is sold
                    Ticket ticket = tickets[i][j];
                    char rowLetter = (char) ('A' + ticket.getRow());
                    int seatNumber = ticket.getSeat() + 1;
                    // Print ticket information
                    System.out.println("Ticket: " + rowLetter + seatNumber +
                            " - Price: €" + ticket.getPrice() +
                            " - Passenger: " + ticket.getPerson().getName() +
                            " " + ticket.getPerson().getSurname() +
                            " - Email: " + ticket.getPerson().getEmail());

                    // Calculate total price
                    totalPrice += ticket.getPrice();
                }
            }
        }

        // Print total price
        System.out.println("Total price of tickets sold during the session: €" + totalPrice);
    }

    private static void searchTickets(Scanner scanner) {
        // Ask the user to input a row letter and seat number
        System.out.println("Enter the row letter (A-D): ");
        char rowLetter = scanner.next().toUpperCase().charAt(0); // Convert input to uppercase
        System.out.println("Enter the seat number (1-14): ");
        int seatNumber = scanner.nextInt();

        // Validate row and seat number
        if (rowLetter < 'A' || rowLetter > 'D' || seatNumber < 1 || seatNumber > 14) {
            System.out.println("Invalid row or seat number.");
            return;
        }

        // Convert row letter to array index
        int row = rowLetter - 'A';

        // Check if the seat is sold
        if (tickets[row][seatNumber - 1] != null) {
            // Seat is sold, print ticket and person information
            Ticket ticket = tickets[row][seatNumber - 1];
            System.out.println("Ticket: " + rowLetter + seatNumber +
                    " - Price: €" + ticket.getPrice() +
                    " - Passenger: " + ticket.getPerson().getName() +
                    " " + ticket.getPerson().getSurname() +
                    " - Email: " + ticket.getPerson().getEmail());
        } else {
            // Seat is available
            System.out.println("This seat is available.");
        }
    }

}
```

```
public class person {
    private String name;
    private String surname;
    private String email;

    // 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 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 information
//    public void printInfo() {
//        System.out.println("Name: " + name);
//        System.out.println("Surname: " + surname);
//        System.out.println("Email: " + email);
//    }
```

```
   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;
    }

    // Constructor, getters, and setters

    public void save() {
        String filename = (char) ('A' + row) + String.valueOf(seat) + ".txt";

        try {
            FileWriter writer = new FileWriter(filename);
            writer.write("Ticket Information\n");
            writer.write("Row: " + (char) ('A' + row) + "\n");
            writer.write("Seat: " + seat + "\n");
            writer.write("Price: €" + price + "\n");
            writer.write("Passenger: " + person.getName() + " " + person.getSurname() + "\n");
            writer.write("Email: " + person.getEmail() + "\n");
            writer.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 print information of a Ticket
//    public void printInfo() {
//        System.out.println("Ticket Information:");
//        System.out.println("Row: " + row);
//        System.out.println("Seat: " + seat);
//        System.out.println("Price: $" + price);
//        System.out.println("Person Information:");
//        person.printInfo(); // Call the printInfo method of Person object
//    }
}
```