Class Flight:

public class Flight {  
 private String FlightNumber;  
 private String[] Cities = {"Dammam", "Jeddah", "Yanbu", "Abha", "Hail", "Tabuk", "Taif"};  
 private int Destination;  
 private int Gate;  
 private String Date;  
 private String DepartureTime;  
 private String ArrivalTime;  
 public static int TotalFlights = 0;  
  
 //Constructors  
 public Flight(){  
 TotalFlights++;  
 FlightNumber ="";  
 Destination =0;  
 Gate = 0;  
 Date = "";  
 DepartureTime="";  
 ArrivalTime="";  
   
 }  
 public Flight(int destination, String date, int gate, String departure){  
 TotalFlights++;  
 Destination=destination;  
 Date=date;  
 Gate=gate;  
 DepartureTime=departure;  
 calculateArrivalTime();  
 generateFlightNumber();  
   
 }  
  
//setters  
 public void setDestination(int destination ) {  
 Destination = destination;  
 }  
 public void setDate(String date){  
 Date=date;  
 }  
 public void setGate(int gate){  
 Gate = gate;  
 }  
 public void setDepartureTime(String dep){  
 DepartureTime = dep;  
 }  
 public void setArrivalTime(String arr){  
 ArrivalTime = arr;  
 }  
  
  
//getters  
 public String getFlightNumber(){  
 return FlightNumber;  
 }  
 public int getDestination(){  
 return Destination;  
 }  
  
 public String getDate(){  
 return Date;  
 }  
 public int getGate(){  
 return Gate;  
 }  
 public String getDepartureTime(){  
 return DepartureTime;  
 }  
 public String getArrivalTime(){  
 return ArrivalTime;  
 }  
  
//generate flight number  
 public void generateFlightNumber(){  
 FlightNumber = Cities[Destination-1].substring(0,3).toUpperCase()+"00"+TotalFlights;  
 }  
  
//calculate arrival time  
 public void calculateArrivalTime(){  
 int hour = Integer.parseInt(DepartureTime.substring(0,DepartureTime.indexOf(':')));  
 int min = Integer.parseInt(DepartureTime.substring(DepartureTime.indexOf(':')+1));  
 hour++;  
 switch(Destination){  
 case 1:   
 min += 5;   
 break;  
 case 2: case 3: case 4:  
 min += 45;   
 break;  
 case 5:  
 min += 15;   
 break;  
 case 6:  
 min += 20;   
 break;  
 case 7:  
 min += 35;  
 }//end switch  
 if (min>=60){  
 hour++;  
 min -= 60;  
 }  
 if (hour>=24){  
 hour -= 24;  
 if (hour<10 && min<10) ArrivalTime= "0"+hour+":0"+min+" +1";  
 else if (hour<10 && min>=10) ArrivalTime= "0"+hour+":"+min+" +1";  
 else if (hour>=10 && min<10) ArrivalTime= hour+":0"+min+" +1";  
 else ArrivalTime= hour+":"+min+" +1";  
 } else{  
 if (hour<10 && min<10) ArrivalTime= "0"+hour+":0"+min;  
 else if (hour<10 && min>=10) ArrivalTime= "0"+hour+":"+min;  
 else if (hour>=10 && min<10) ArrivalTime= hour+":0"+min;  
 else ArrivalTime= hour+":"+min;  
 }   
 }  
  
//print flight info  
 public void printFlightInfo(){  
 System.out.printf("Flight Number: %-15s Gate: %-15d %n",FlightNumber, Gate );  
 System.out.printf("Destination: %-15s Date: %-15s %n",Cities[Destination-1],Date);  
 System.out.printf("Depature Time: %-15s Arrival Time: %-15s %n", DepartureTime,ArrivalTime);  
 System.out.println();  
 }  
  
  
}

# Class Airline:

import java.util.Scanner;  
public class Airline {  
 static Scanner input = new Scanner (System.in);  
 private static Flight[] FlightsList = new Flight[100];  
 static String error = "Operation is not successful, make sure to enter the informaion in the right format.\n";  
 public static void main (String[] args){  
 int x, destination, gate;  
 String date, departure, fn;  
 do{  
 System.out.println("What would you like to do?\n(1) Add a Flight\n(2) Find a flight\n(3) List of all flights\n(4) List of flights for a given date\n(5) Update Departure & Arrival Time\n(6) Total number of flights\n(7) Exit");  
 x = input.nextInt();  
 System.out.println();  
 switch (x){  
 case 1:   
   
 System.out.println("Choose flight's destinaion\n(1) Dammam\n(2) Jeddah\n(3) Yanbu\n(4) Abha\n(5) Hail\n(6) Tabuk\n(7) Taif");  
 destination = input.nextInt();  
 System.out.println();  
 System.out.print("Date in the format (dd/mm): ");  
 date = input.next();  
 System.out.print("Departure time in the format (hh:mm) and 24-hours system: ");  
 departure = input.next();  
 System.out.print("Gate: ");  
 gate = input.nextInt();  
 if (!addFlight (destination, date, gate, departure)){  
 System.out.println(error);  
 }  
 else System.out.println("Flight added successfully");  
 System.out.println();  
 break;  
 case 2:  
 System.out.println("Enter flight number:");  
 fn = input.next();  
 if (findFlight(fn)!= -1) System.out.println("The index of this flight is "+findFlight(fn));  
 else System.out.println("The flight was not found.");  
 System.out.println();  
 break;  
 case 3:   
 printAll();  
 System.out.println();  
 break;  
 case 4:  
 System.out.println("Enter the date:");  
 String d = input.next();  
 printAll(d);  
 System.out.println();  
 break;  
 case 5:  
 System.out.println("Enter flight number:");  
 fn = input.next();  
 System.out.println("Enter new departure time:");  
 d = input.next();  
 updateTime(fn,d);  
 System.out.println();  
 break;  
 case 6:  
 System.out.println("Total number of flights: "+getNumberOfFlights());  
 System.out.println();  
 break;  
 case 7:   
 break;  
 default: System.out.print("Invalid input");  
 } // end switch  
   
 } while (x != 7); // end do while loop  
 System.out.print("End of program");  
 } // end main  
  
 public static boolean addFlight(int destination,String date,int gate,String departure){  
 if (Flight.TotalFlights>=100)  
 return false;  
 if (destination>7||destination<1)  
 return false;  
 if(departure.length()!=5||departure.charAt(2)!=':')  
 return false;  
 int hour=Integer.parseInt(departure.substring(0,2));  
 int min=Integer.parseInt(departure.substring(3));  
 if (hour>23||hour<0)  
 return false;  
 if (min<0||min>60)  
 return false;  
 if(date.charAt(2)!='/'||date.length()!=5)  
 return false;  
 int day=Integer.parseInt(date.substring(0,2));  
 if(day<1||day>31)  
 return false;  
 int month=Integer.parseInt(date.substring(3));  
 if(month<1||month>12)  
 return false;  
   
 for(int i=0;i<Flight.TotalFlights;i++){  
 if(FlightsList[i].getDestination()==destination  
 && FlightsList[i].getGate()==gate  
 && FlightsList[i].getDate().equals(date)  
 && FlightsList[i].getDepartureTime().equals(departure))   
 return false;  
 }  
   
 FlightsList[Flight.TotalFlights]= new Flight(destination, date, gate,departure);   
 return true;  
 }  
   
   
 public static int findFlight(String FlightNumber){  
   
 for (int i=0; i<Flight.TotalFlights;i++)  
 if (FlightsList[i].getFlightNumber().equals(FlightNumber))  
 return i;  
   
 return -1;  
 }   
   
 public static void printAll(){  
 for (int i=0; i<Flight.TotalFlights; i++)  
 FlightsList[i].printFlightInfo();  
 }  
   
 public static void printAll (String date){  
 for (int i=0;i<Flight.TotalFlights;i++){  
 if (FlightsList[i].getDate().equals(date))  
 FlightsList[i].printFlightInfo();  
 }  
   
 }  
   
 public static void updateTime( String FlightNumber,String departure){  
 if(departure.length()!=5||departure.charAt(2)!=':'){  
 System.out.println(error);  
 return ;  
 }  
 int hour=Integer.parseInt(departure.substring(0,2));  
 int min=Integer.parseInt(departure.substring(3));  
 if (hour>23||hour<0||min<0||min>60){  
 System.out.println(error);  
 return ;  
 }  
 int i;  
 for(i=0;i<Flight.TotalFlights;i++)  
 if((FlightsList[i].getFlightNumber()).equals(FlightNumber))  
 break;  
 if((FlightsList[i].getFlightNumber()).equals(FlightNumber)){  
 FlightsList[i].setDepartureTime(departure);  
 FlightsList[i].calculateArrivalTime();  
 FlightsList[i].printFlightInfo();  
 } else System.out.println("The flight was not found.\n");  
   
 }  
 public static int getNumberOfFlights (){  
 return Flight.TotalFlights;  
   
 }  
  
}

# Sample rum

## Option 1: Add a flight

### Add flight number 1:

A screenshot of a cell phone

Description generated with very high confidence

### Add flight number 2:

A screenshot of a cell phone

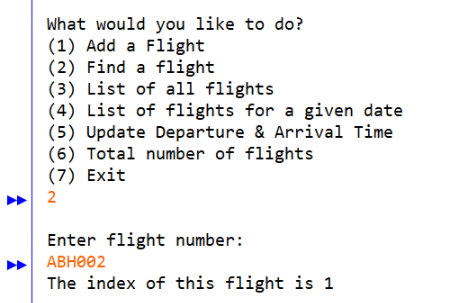
Description generated with high confidence

### Add flight number 3:

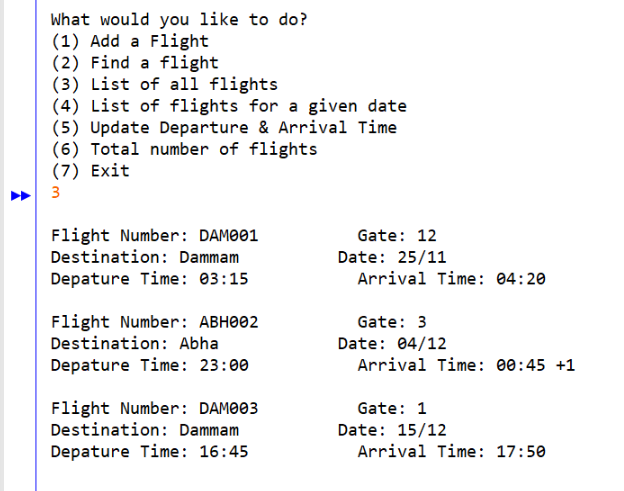
A screenshot of a cell phone

Description generated with very high confidence

## Option 2: Find a flight



## Option 3: List of all flights



## Option 4: List of flights for a given date

A screenshot of a cell phone

Description generated with very high confidence

## Option 5: Update departure and arrival time

A screenshot of a cell phone

Description generated with very high confidence

## Option 6&7: Total number of flights & Exit

A screenshot of a cell phone

Description generated with very high confidence