COMPUTER

PROJECT

NAME: SHRIVARDHAN GOENKA

CLASS: 11 S1

ROLL NUMBER: 12967

HOUSE: HASTINGS

INDEX

INTRODUCTION

This is a flight reservation system which can book a flight for upto 10 passengers between any 2 cities in a list of 10 in the country. The Airline company name is fictitious which is DomesticFlights.

There are 136 direct flights daily (multiple flights daily between common locations) operating and some non-direct flights between cities that are not commonly travelled to and fro from. However, there is atleast 1 flight between all the 10 locations per day. The prices range between Rs3000 and Rs 14000. The schedule and prices of the flights are derived from the AirIndia schedule and prices on the 1st of January, 2021.

The system has a menu driven interface (CUI- Command User Interface) which lays out the options and asks the user for a option. The program handles all exceptions and ideally there should not be any circumstances in which the program will crash due to unregulated input from the user.

The program has 5 options on the main page which are: Book a new flight, check a previous booking, to see the network of flights, to read the 'About Us' section and to quit.

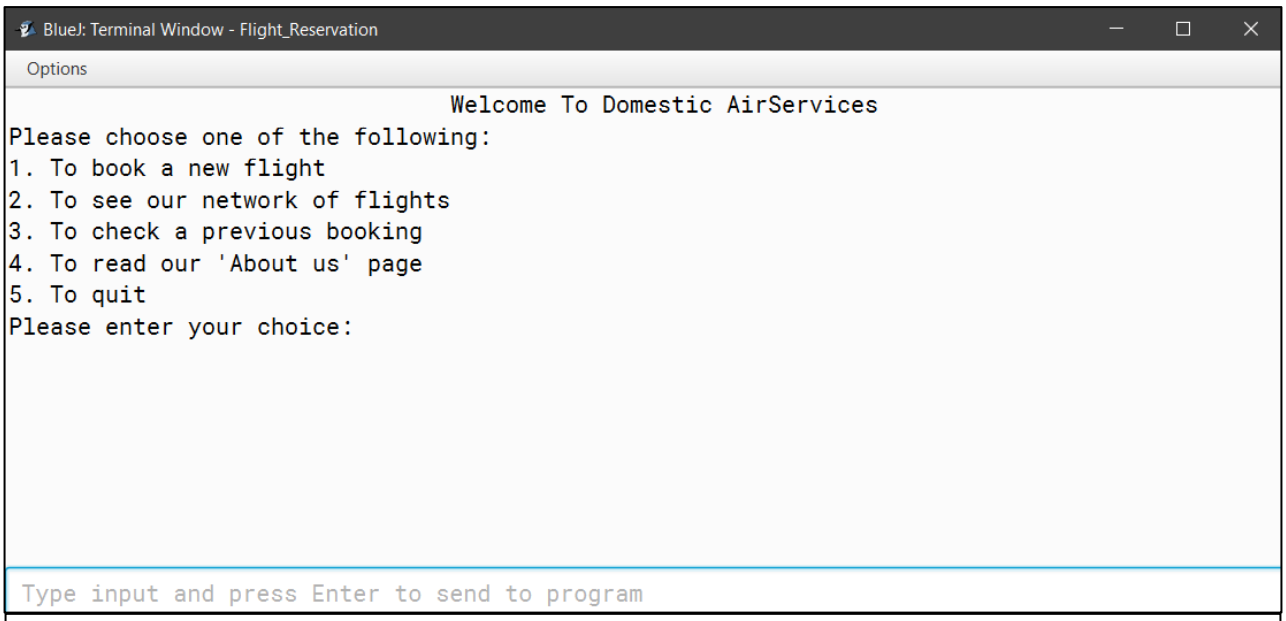
A user manual is given after this, which by using screenshots explains the entire process of how to use this program.

All the bookings made are stored permanently as files on the disk. There are two types of files which are edited by the program. The first type of file is the one which stores the seats booked for all the flights for each day and the other type is the one which stores all the previous bookings by the name of the passenger codes.

The other files are prewritten and are used to display certain information such as the entire schedule. These files are present in the pdf later on.

User Manual

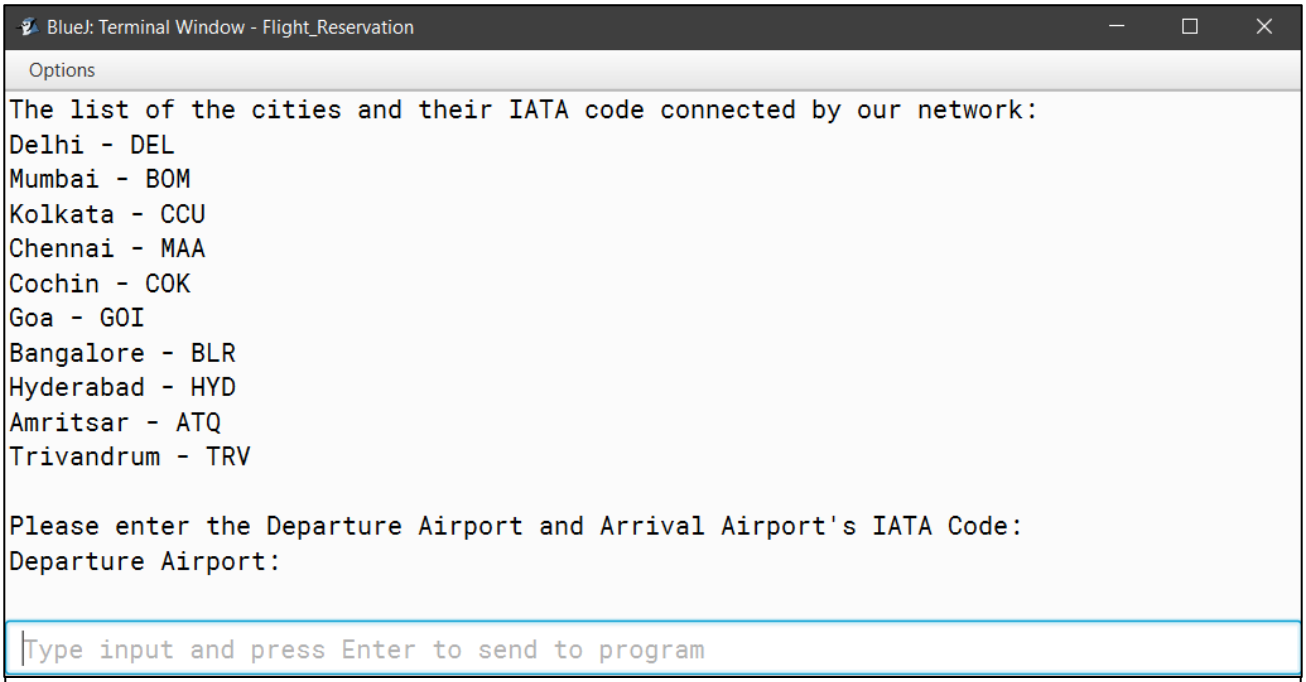
To run the program, please call the main function in the class: Main



^ Main Page/ Home Page

Please choose any one of the options. Please don't worry if you select the wrong option. There are options given in all pages to return back to this menu.
However, to the quit, the easiest way is to use the command: Ctr+W if you are not on the main page.

Option 1: Book a new flight

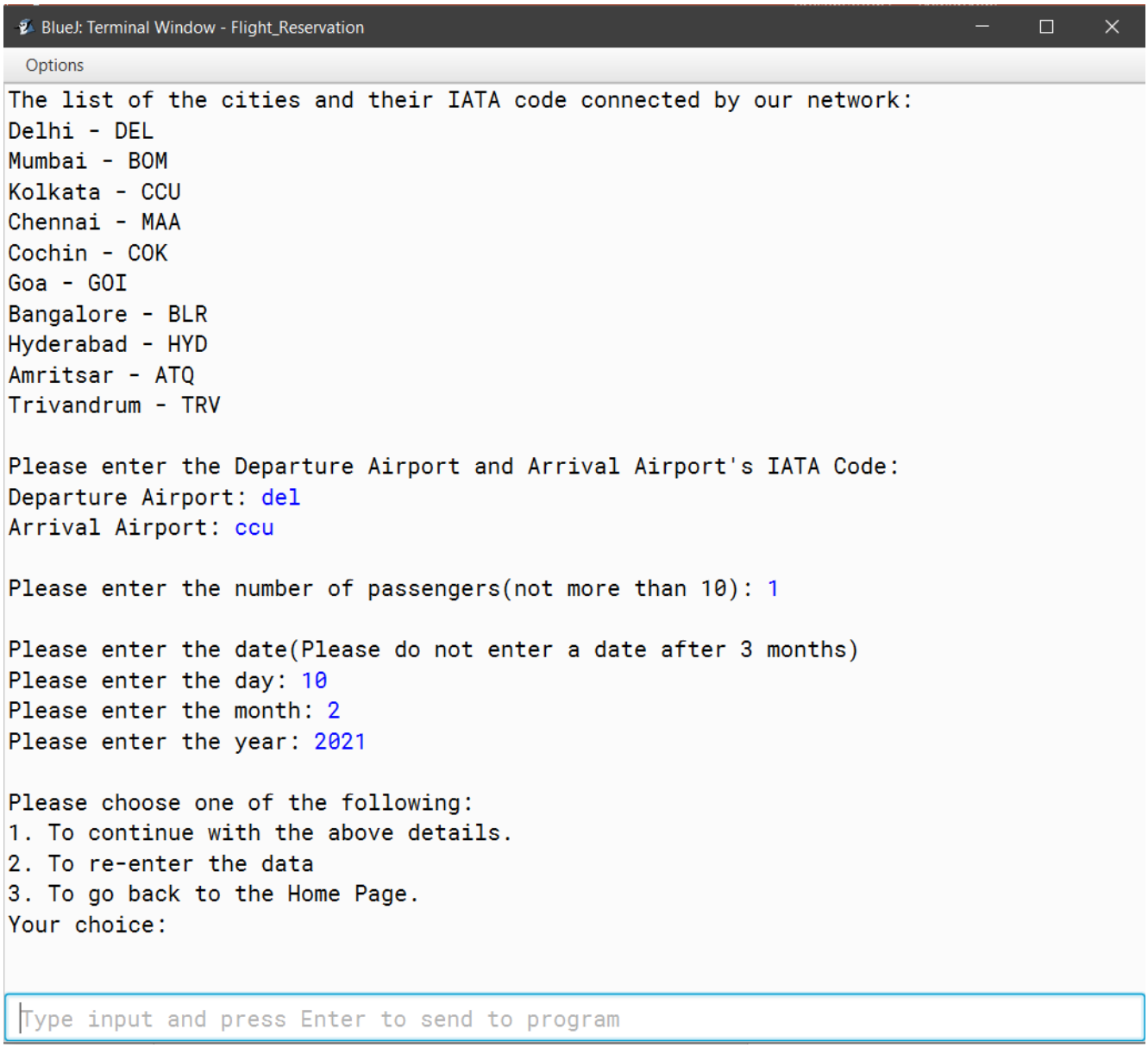


This is the first page to book a new flight. Please enter the details as asked in the program. At the end, there will be an option to reset the page if you have made any mistake.

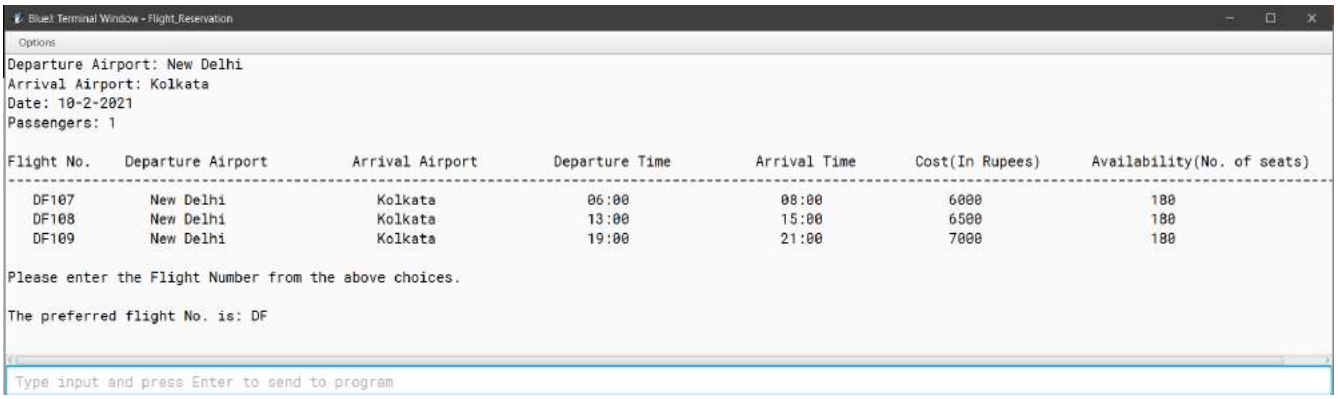
Some important points to be noted while entering the information in this page:

- It is not case sensitive, so the user can enter the data in any case.
- The number of passengers should be not be more than 10.
- The date of travel should be from one day after the date of usage to 3 months from them. Bookings cannot be made for that day itself and cannot be made for more than 3 months from then.

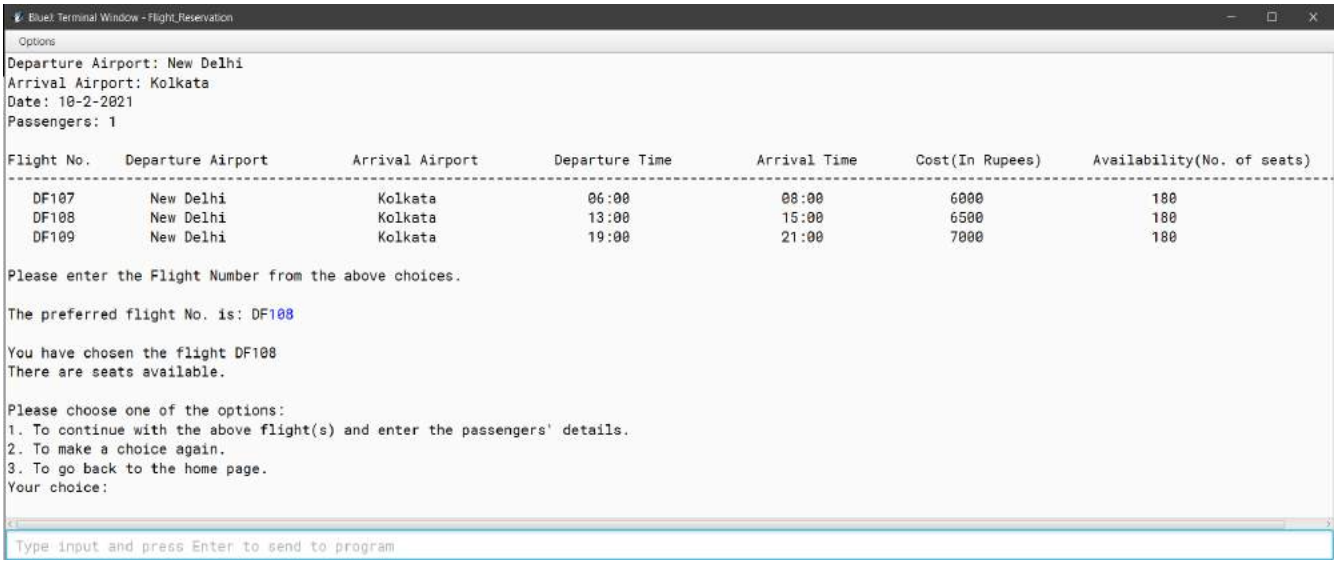
Failure to follow these points will give an error and the user will have to re-enter the data.



This is the page filled up. All the rules have been followed in this page and thus there was no error.
Now the user has 3 options as given in the picture above. The user may choose any one of them.



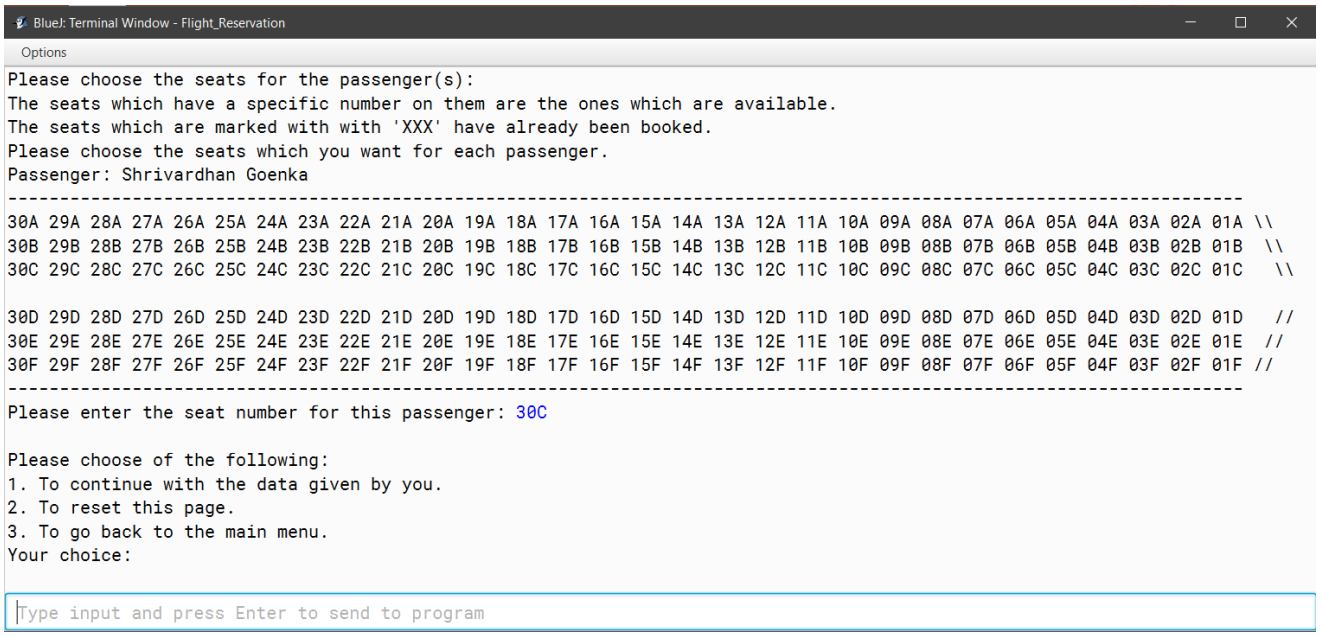
This is the second page of the Book a new flight option which can be reached by choosing option 1 on the last page.
The user may now choose any option(flight as desired). This is for the direct flights. The page for indirect flights is slightly different but the basic concept is the same.
The user may choose any option as per his wishes.



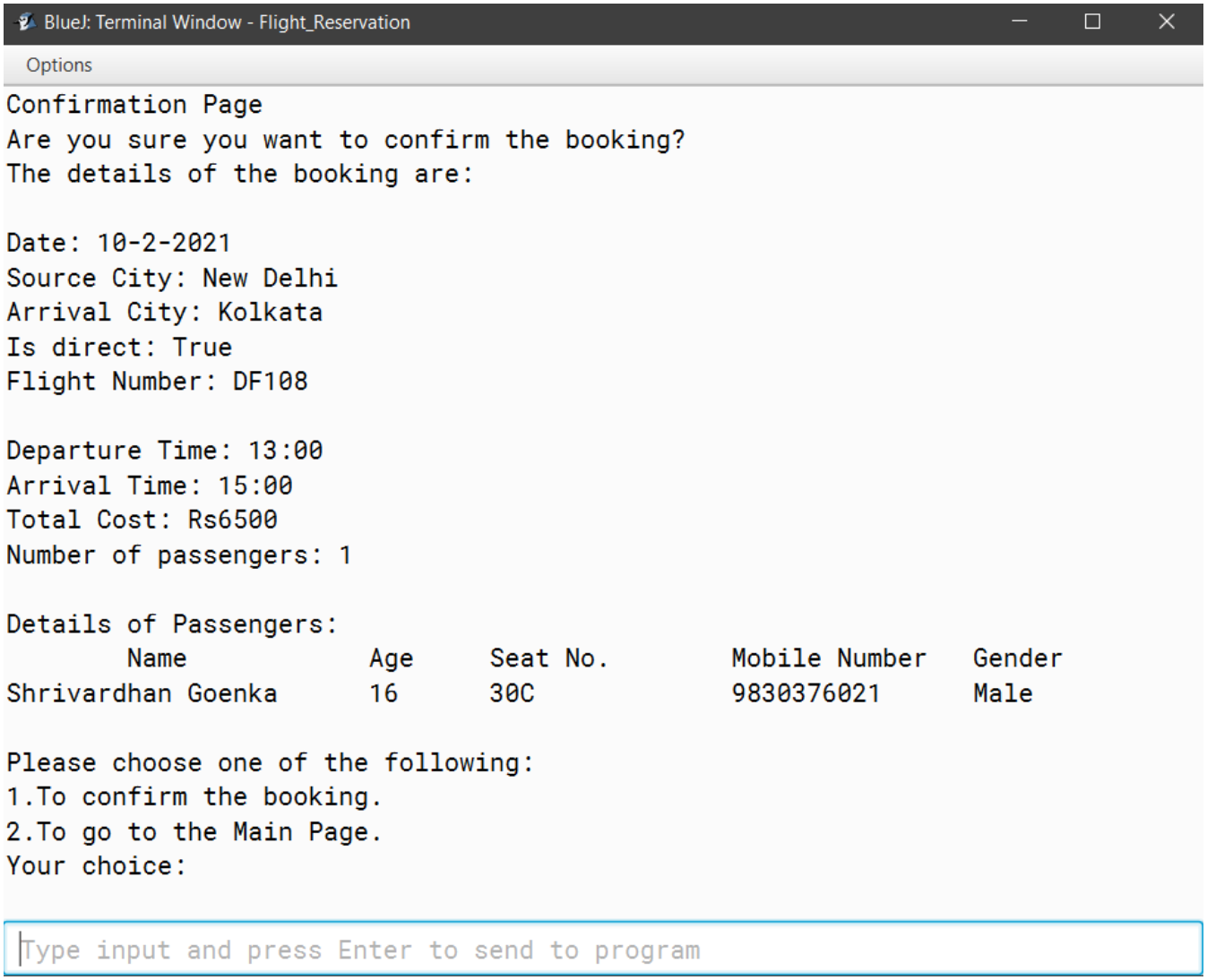
This is the second page which is filled up.
The user again has 3 options which are similar to that of the last page.

Please note: The clarity is bad due to the large size of the window.

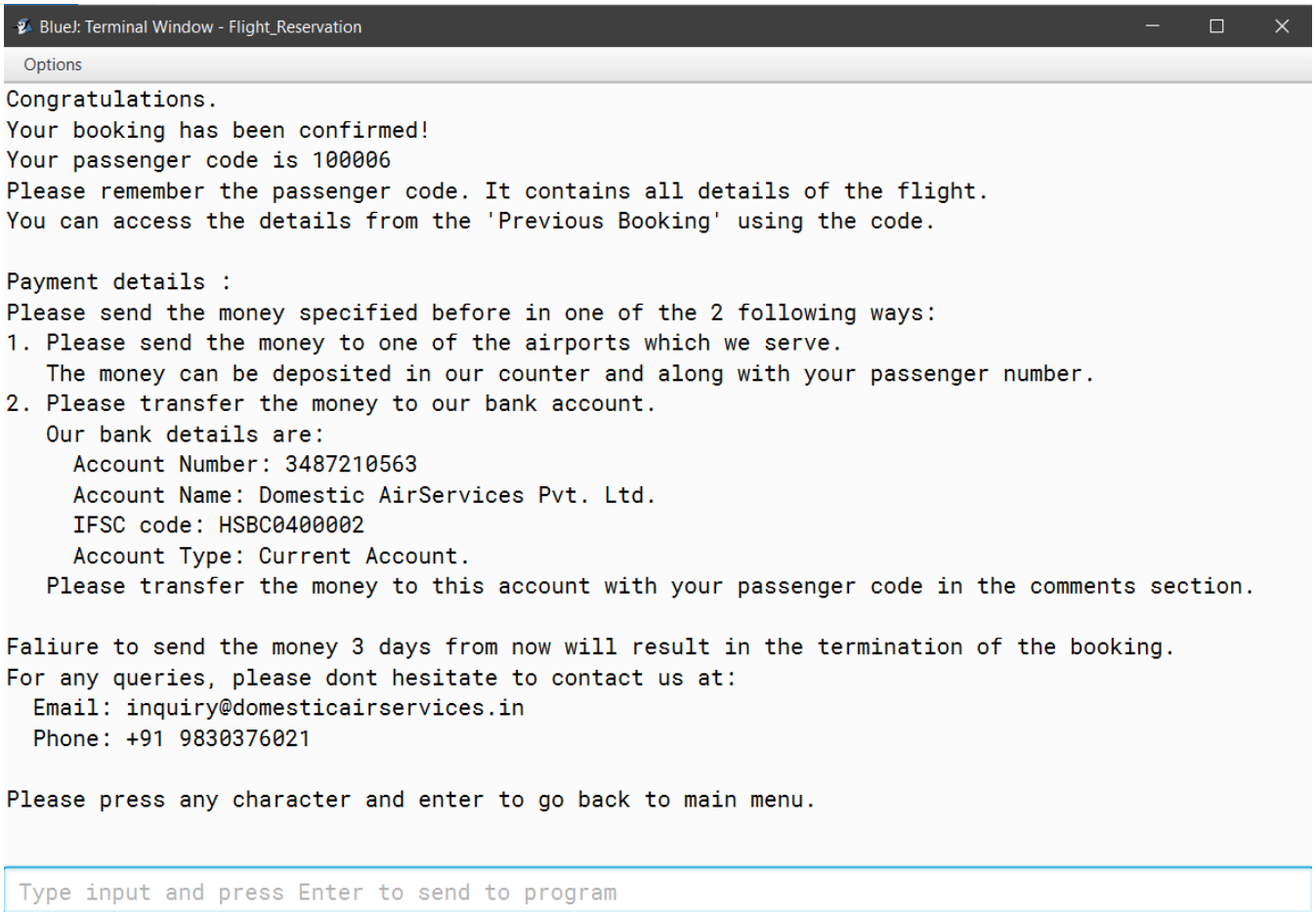
This is the third page which asks for the information of the passenger(s). The form above is already filled and the user may also do so by following the instructions given.



This is the fourth page which asks for the seat number the passenger(s) wants to sit on. The seats which are not available will be marked by: ‘XXX’.
The user may choose any seat as he desires, provided it is empty.



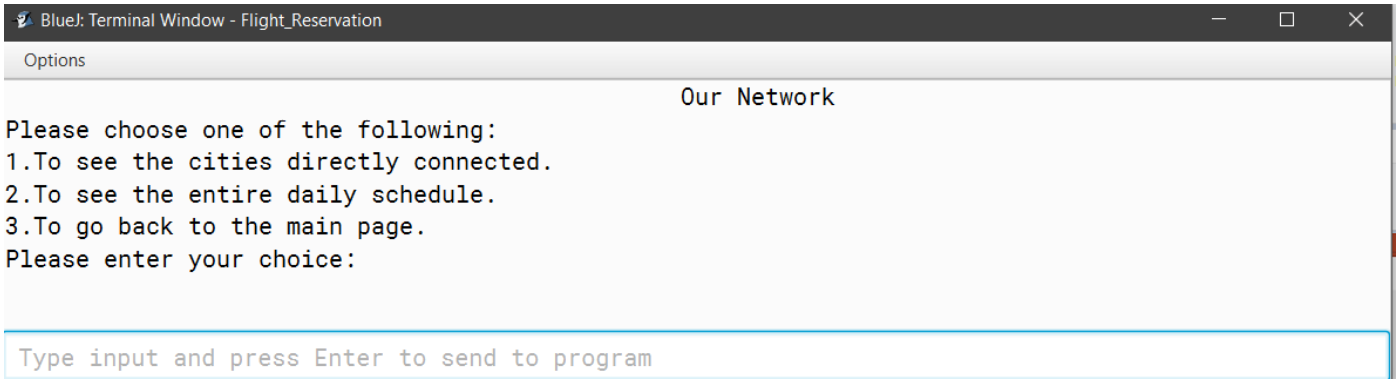
This is the last and final part of the booking section which is the confirmation page. The user needs to go through the details and confirm that all of them are correct. If yes, then the user can press 1 and confirm the booking which in turn will save the booking on the system. If any detail is incorrect, the user needs to press 1 and start all over. For this reason, the system has given the option to reset the page on every single step/page.
Once Confirmed, there is no option to edit or cancel the booking.



This is the page that confirms the booking. The passenger code is also provided which the user needs to remember to access the details of the bookings at a later stage. To continue and go back to the main menu, the user needs to press any key and press enter.
Please note: All the bank details and contact details given are fake.

Option 2: To check the network of flights

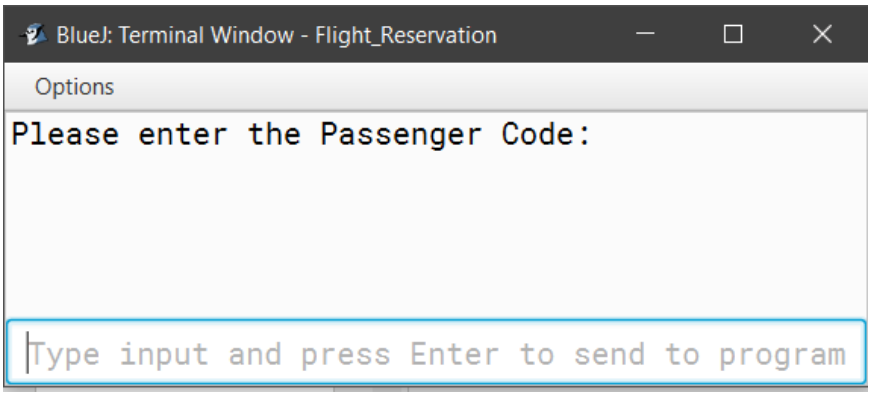
This option is to check all the flights the company operates.



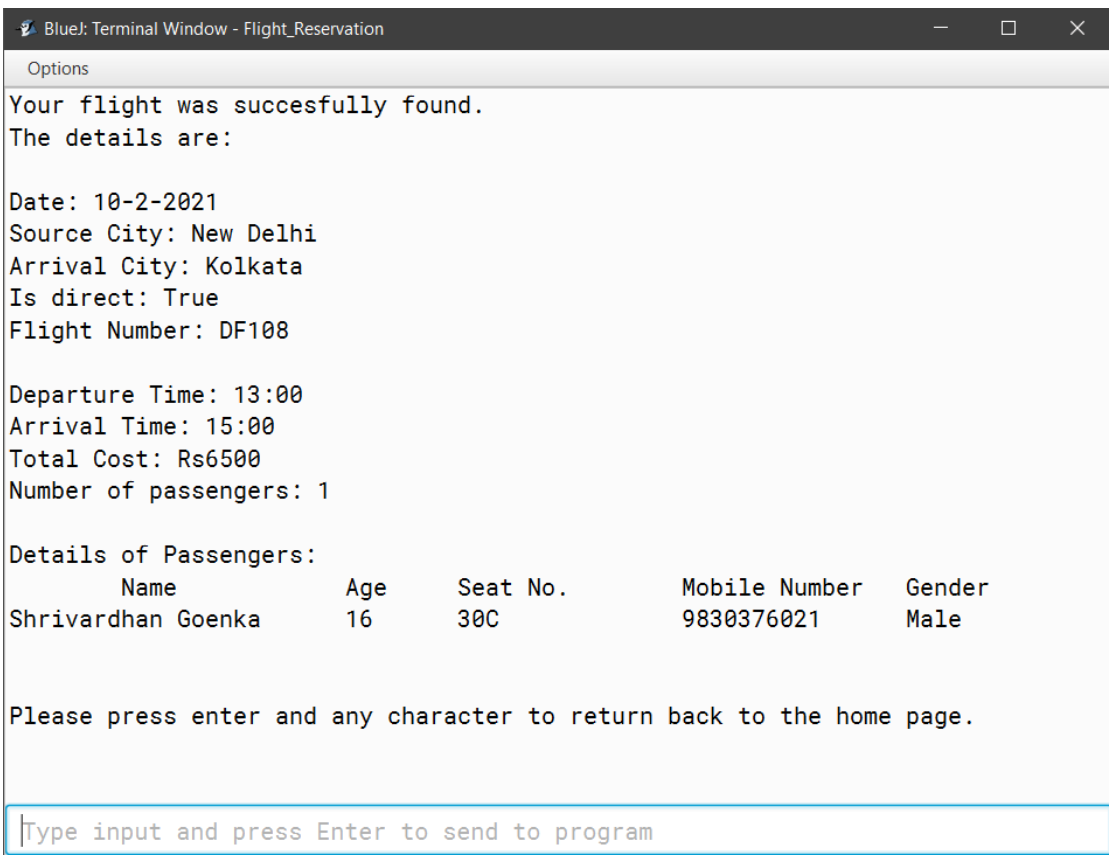
This is the first page. The user may choose any of the options.
Due to the large size of the data, these options are not visited in this manual but are present in the Files Section later in the PDF.

Option 3: To check a previous booking

This option is to check a booking previously made.

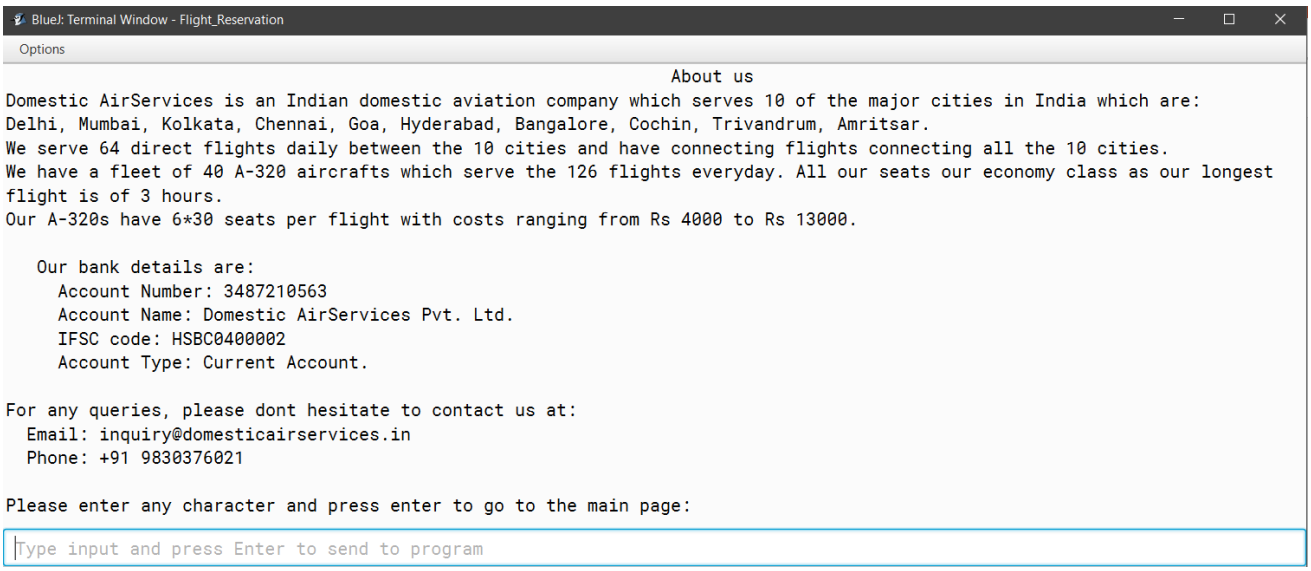


This is the first page. The user needs to enter the correct code to check any previous bookings.
On entering 100006 which was the generated code for the booking shown in this manual, the following is obtained:

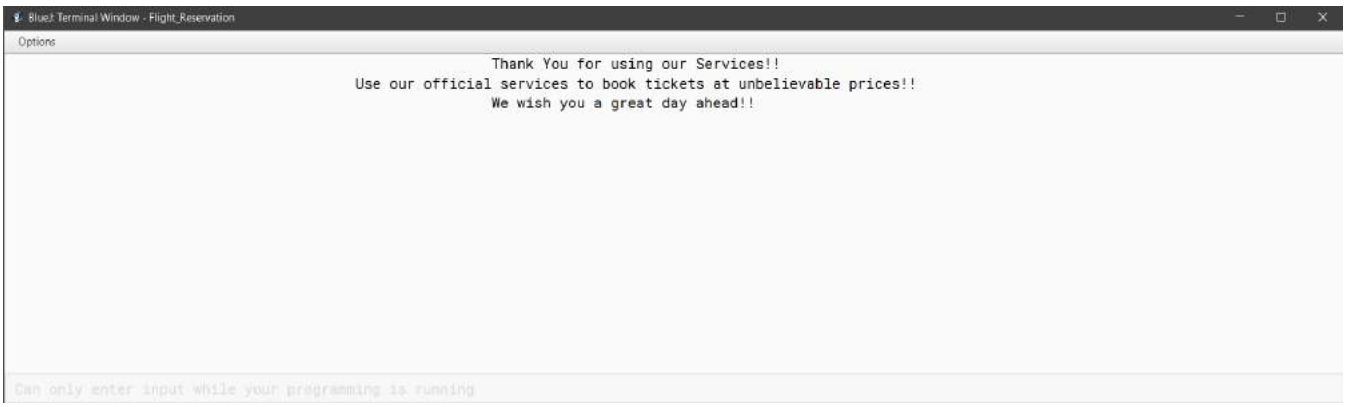


This shows the details of the flight. To continue and go back to the main menu you need to press any key and then enter.
Note: In case a correct code is not entered, a message will be shown and the user can try again.

Option 4: To read “About us”

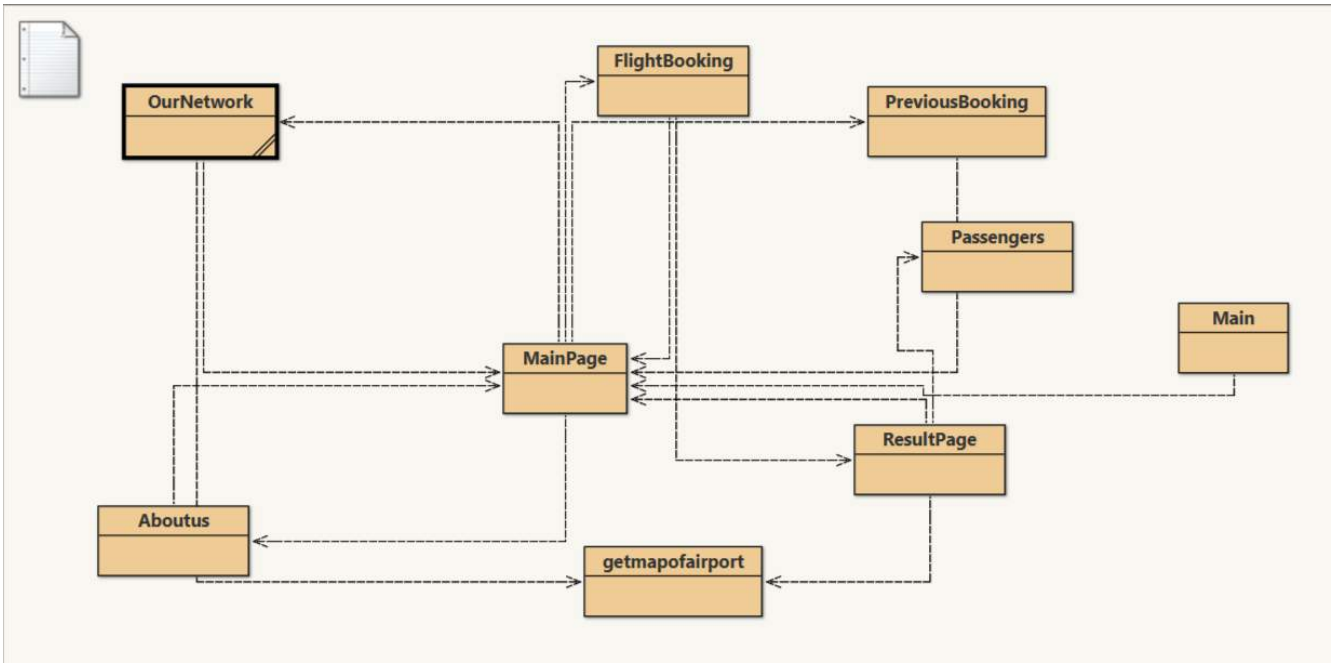


Option 5: To quit



Code for the System

Class Diagram



Order of classes:

The following order has been followed in the PDF:

- Main
- MainPage
- FlightBooking
- OurNetwork
- Aboutus
- Getmapofairport
- Previous Booking
- ResultPage
- Passengers

```
1 class Main
2 {
3     public static void main()
4     {
5         MainPage mp = new MainPage();
6         mp.mainpage();
7     }
8 }
```

```
1  import java.util.Scanner;
2  import java.io.*;
3  class MainPage
4  {
5      static BufferedReader brline = new BufferedReader(new
   InputStreamReader(System.in));
6
7      public static void mainpage()
8      {
9          System.out.println("\f\t\t\t\tWelcome To Domestic AirServices");
10         System.out.println("Please choose one of the following: ");
11         System.out.println("1. To book a new flight");
12         System.out.println("2. To see our network of flights");
13         System.out.println("3. To check a previous booking");
14         System.out.println("4. To read our 'About us' page");
15         System.out.println("5. To quit");
16         choicemainpage();
17     }
18     static int c;
19     private static void choicemainpage()
20     {
21         Scanner sc = new Scanner(System.in);
22         System.out.print("Please enter your choice: ");
23         try
24         {
25             c = Integer.parseInt(brline.readLine());
26         }
27         catch(Exception e)
28         {
29             System.out.println("Wrong choice. Please enter an integer.");
30             choicemainpage();
31         }
32         switch(c)
33         {
34             case 1:
35             FlightBooking.mainpage();
36             break;
37             case 2:
38             OurNetwork.mainpage();
39             break;
40             case 3:
41             System.out.print("\f");
42             PreviousBooking.mainpage();
43             break;
44             case 4:
45             Aboutus.mainpage();
46             mainpage();
47             break;
48             case 5:
49             quit();
```

```
            break;
            default:
            System.out.println("Wrong choice. Please enter again.");
            choicemainpage();
            break;
        }
    }

    private static void quit()
    {
        System.out.println("\f\t\t\t\t\t\t\t Thank You for using our Services!!");
        System.out.println("\t\t\t\t\t Use our official services to book tickets at
unbelievable prices!!");
        System.out.println("\t\t\t\t\t\t\t We wish you a great day ahead!!");
        System.exit(0);
    }
}
```

```
1  import java.util.Scanner;
2  import java.io.*;
3  import java.time.format.DateTimeFormatter;
4  import java.time.LocalDateTime;
5  class FlightBooking
6  {
7      static String dep;
8      static String arr;
9      static int pas;
10     static String date;
11     static BufferedReader brline = new BufferedReader(new
   InputStreamReader(System.in));
12     public static void mainpage()
13     {
14         Scanner sc = new Scanner(System.in);
15         System.out.println("\fThe list of the cities and their IATA code connected
   by our network: \n"+
16             "Delhi - DEL\n"+
17             "Mumbai - BOM\n"+
18             "Kolkata - CCU\n"+
19             "Chennai - MAA\n"+
20             "Cochin - COK\n"+
21             "Goa - GOI\n"+
22             "Bangalore - BLR\n"+
23             "Hyderabad - HYD\n"+
24             "Amritsar - ATQ\n"+
25             "Trivandrum - TRV");
26         System.out.println("\nPlease enter the Departure Airport and Arrival
   Airport's IATA Code: ");
27         dep = enterdeparr("Departure Airport");
28         dep = dep.toUpperCase();
29         arr = enterdeparr("Arrival Airport");
30         arr = arr.toUpperCase();
31         if(dep.equalsIgnoreCase(arr))
32         {
33             for(;;)
34             {
35                 System.out.println("Departure and Arrival Airport cannot be the
   same");
36                 arr= enterdeparr("Arrival Airport");
37                 if(arr.equalsIgnoreCase(dep)==false)
38                 break;
39             }
40         }
41         System.out.println();
42         pas=passengers();
43         System.out.println();
44         System.out.println("Please enter the date(Please do not enter a date after 3
   months)");
45         date = enterdate();
```

```
46
47          System.out.println("\nPlease choose one of the following: ");
48          System.out.println("1. To continue with the above details.\n"+
49                             "2. To re-enter the data\n"+
50                             "3. To go back to the Home Page.");
51          checkcontinue();
52      }
53
54      private static boolean checkIATAcode(String str)
55      {
56          try
57          {
58              BufferedReader br = new BufferedReader(new
    FileReader("airportscode.txt"));
59              String text;
60              while((text=br.readLine())!=null)
61              {
62                  if(str.equalsIgnoreCase(text))
63                      return true;
64              }
65              return false;
66          }
67          catch(IOException e)
68          {
69
70          }
71          return false;
72      }
73
74      private static String enterdeparr(String str)
75      {
76          Scanner sc = new Scanner(System.in);
77
78          System.out.print(str+": ");
79          String dep = sc.next();
80          if(checkIATAcode(dep))
81              return dep;
82          else
83          {
84              System.out.println("Wrong code. Please try again.");
85              return enterdeparr(str);
86          }
87      }
88      static int d,m,y;
89       static String enterdate()
90      {
91          Scanner sc = new Scanner(System.in);
92          String str;
93          try
94          {
```

```
95              System.out.print("Please enter the day: ");
96              d = Integer.parseInt(brline.readLine());
97              System.out.print("Please enter the month: ");
98              m = Integer.parseInt(brline.readLine());
99              System.out.print("Please enter the year: ");
100             y = Integer.parseInt(brline.readLine());
101         }
102         catch(Exception e)
103         {
104             System.out.println("\nPlease enter integers only.");
105             enterdate();
106         }
107         if(m>12 ||d<1||
y<2020||y>2022||((m==1||m==3||m==5||m==7||m==8||m==10||m==12)&&d>31)||((m==2)&&d>28)
||((m==4||m==6||m==9||m==11)&&d>30))
108         {
109             System.out.println("Wrong date entered. Please try again.\n");
110             return enterdate();
111         }
112         else if(checkdate(d,m,y)==false)
113         {
114             System.out.println("Please enter a date from tomorrow to 3 months from
now. Please try again.\n");
115             return enterdate();
116         }
117         str = d+"-"+m+"-"+y;
118         return str;
119     }
120     private static boolean checkdate(int d,int m,int y)
121     {
122         DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MM/yyyy");
123         LocalDateTime now = LocalDateTime.now();
124         String date = dtf.format(now);
125         int dc = Integer.parseInt(date.substring(0,2));
126         int mc = Integer.parseInt(date.substring(3,5));
127         int yc = Integer.parseInt(date.substring(6));
128         int ym=yc;
129         int mm;
130         mm=mc+3;
131         if(mm>=13)
132         {
133             mm-=12;
134             ym++;
135         }
136
if(y>ym||y<yc||(y==yc&&m==mc&&d<=dc)||(y==ym&&m==mm&&d>dc)||(y==yc&&m<mc)||(y==ym&&m
>mm))
137         return false;
138         return true;
139     }
```

```
140     static int c=0;
141     private static void checkcontinue()
142     {
143         Scanner sc = new Scanner(System.in);
144         System.out.print("Your choice: ");
145         try
146         {
147             c = Integer.parseInt(brline.readLine());
148         }
149         catch(Exception e)
150         {
151             System.out.println("\nPlease enter an integer.");
152             checkcontinue();
153         }
154         switch(c)
155         {
156             case 1:
157             ResultPage.enter(dep,arr,pas,date);
158             break;
159             case 2:
160             mainpage();
161             break;
162             case 3:
163             MainPage.mainpage();
164             break;
165             default:
166             System.out.println("Wrong choice entered. Please try again.");
167             checkcontinue();
168         }
169     }
170     static int p=0;
171     private static int passengers()
172     {
173         Scanner sc = new Scanner(System.in);
174         System.out.print("Please enter the number of passengers(not more than 10):
    ");
175         try
176         {
177             p = Integer.parseInt(brline.readLine());
178         }
179         catch(Exception e)
180         {
181             System.out.println("\nPlease enter an integer only.");
182             passengers();
183         }
184         if(p>0&&p<=10)
185         return p;
186         System.out.println("The number of passengers is incorrect. Please try
    again.");
187         return passengers();
```

```
188         }
189 }
```

```java
import java.util.Scanner;
import java.io.*;
import java.util.Map;
class OurNetwork
{
    static int c=0;
    static BufferedReader brline = new BufferedReader(new
InputStreamReader(System.in));
    public static void mainpage()
    {
        System.out.println("\f\t\t\t\t\t\tOur Network");
        System.out.println("Please choose one of the following:\n"+
            "1.To see the cities directly connected.\n"+
            "2.To see the entire daily schedule.\n"+
            "3.To go back to the main page.");
        choicemainpage();
    }

    private static void choicemainpage()
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Please enter your choice: ");
        try
        {
            c = Integer.parseInt(brline.readLine());
        }
        catch(Exception e)
        {
            System.out.println("Please enter an integer.");
            choicemainpage();
        }
        switch(c)
        {
            case 1:
            dirconnected();
            break;
            case 2:
            schedule();
            break;
            case 3:
            MainPage.mainpage();
            break;
            default:
            System.out.println("Wrong choice. Please enter again.");
            choicemainpage();
            break;
        }
    }

    private static void dirconnected()
```

```
50      {
51          try
52          {
53              Map <String,String> map = getmapofairport.main();
54              BufferedReader br = new BufferedReader(new
FileReader("DirectFlights.txt"));
55              System.out.println("\nDeparture Airport \tArrival Airport \t No of
flights per day");
56
System.out.println("-----------------------------------------------------------------
-------");
57              String text;
58              while((text=br.readLine())!=null)
59              {
60                  System.out.println("    "+map.get(text.substring(0,3))+ "\t\t
"+map.get(text.substring(4,7))+"\t\t\t"+text.charAt(8));
61              }
62              printstatement();
63          }
64          catch(IOException e)
65          {
66
67          }
68      }
69      private static void printstatement()
70      {
71          System.out.println("\n\nPlease choose one of the following: \n"+
72                             "1. To go back to the main menu.\n"+
73                             "2. To reset this page.");
74          choicesecondpage();
75      }
76      private static void schedule()
77      {
78          try
79          {
80              Map <String,String> map = getmapofairport.main();
81              BufferedReader br = new BufferedReader(new
FileReader("flightlist1.txt"));
82              System.out.println("\nFlight No.    Departure Airport \tArrival Airport
\t Departure Time \t Arrival Time \t    Cost(In Rupees)");
83
System.out.println("-----------------------------------------------------------------
-----------------------------------------------------------");
84              String text;
85              while((text=br.readLine())!=null)
86              {
87                  System.out.println("  DF"+text.substring(0,3)+"\t\t
"+map.get(text.substring(4,7))+"\t\t    "+map.get(text.substring(8,11))+"\t\t     "+
88                      text.substring(12,18)+"\t\t
"+text.substring(18,24)+"\t\t"+text.substring(24));
```

```
89              }
90              printstatement();
91          }
92          catch(IOException e)
93          {
94              schedule();
95          }
96      }
97      private static void choicesecondpage()
98      {
99          Scanner sc = new Scanner(System.in);
100         System.out.print("Please enter your choice: ");
101         try
102         {
103             c = Integer.parseInt(brline.readLine());
104         }
105         catch(Exception e)
106         {
107             System.out.println("Please enter an integer.");
108             choicesecondpage();
109         }
110         switch(c)
111         {
112             case 1:
113             System.out.print("\f");
114             MainPage.mainpage();
115             break;
116             case 2:
117             System.out.println("\f");
118             mainpage();
119             break;
120             default:
121             System.out.println("Wrong choice. Please enter again.");
122             choicemainpage();
123             break;
124         }
125     }
126 }
```

```java
import java.util.Scanner;
class Aboutus
{
    public static void mainpage()
    {
        System.out.println("\f\t\t\t\t\t\t\t\tAbout us");
        System.out.print("Domestic AirServices is an Indian domestic aviation company which serves 10 of the major cities in India which are:\n"+
                         "Delhi, Mumbai, Kolkata, Chennai, Goa, Hyderabad, Bangalore, Cochin, Trivandrum, Amritsar.\n"+
                         "We serve 64 direct flights daily between the 10 cities and have connecting flights connecting all the 10 cities.\n"+
                         "We have a fleet of 40 A-320 aircrafts which serve the 126 flights everyday. All our seats our economy class as our longest \nflight is of 3 hours.\n"+
                         "Our A-320s have 6*30 seats per flight with costs ranging from Rs 4000 to Rs 13000.\n\n"+
                         "   Our bank details are:\n"+
                          "     Account Number: 3487210563\n"+
                          "     Account Name: Domestic AirServices Pvt. Ltd.\n"+
                          "     IFSC code: HSBC0400002\n"+
                          "     Account Type: Current Account.\n\n"+
                          "For any queries, please dont hesitate to contact us at:\n"+
                          "  Email: inquiry@domesticairservices.in\n"+
                          "  Phone: +91 9830376021\n\n"+
                         "Please enter any character and press enter to go to the main page: ");
        Scanner sc = new Scanner(System.in);
        sc.next();
        System.out.print("\f");
        MainPage.mainpage();
    }
}
```

```java
import java.util.Map;
import java.util.HashMap;
import java.io.*;
class getmapofairport
{
    public static Map<String,String> main() throws IOException
    {
        Map<String,String> map=new HashMap<String,String>();
        BufferedReader br = new BufferedReader(new FileReader("airportscode.txt"));
        BufferedReader bw = new BufferedReader(new FileReader("airportsname.txt"));
        String text;
        while((text=br.readLine())!=null)
        {
            map.put(text,bw.readLine());
        }
        return map;
    }
}
```

```
1  import java.util.Scanner;
2  import java.io.*;
3  class PreviousBooking
4  {
5      static int n=0;
6      public static void mainpage()
7      {
8          Scanner sc = new Scanner(System.in);
9
10         System.out.print("Please enter the Passenger Code: ");
11         try
12         {
13             BufferedReader brline = new BufferedReader(new
   InputStreamReader(System.in));
14             n = Integer.parseInt(brline.readLine());
15             try
16             {
17                 BufferedReader br = new BufferedReader(new
   FileReader("Passengers\\Pas"+n+".txt"));
18                 System.out.println("\fYour flight was succesfully found.");
19                 System.out.println("The details are: \n");
20                 String text;
21                 while((text=br.readLine())!=null)
22                 {
23                     System.out.println(text);
24
25                 br.close();
26                 System.out.println("\n\nPlease press enter and any character to
   return back to the home page.");
27                 sc.next();
28                 MainPage.mainpage();
29             }
30             catch(Exception e)
31             {
32                 System.out.println("\nThe code is not valid. Please try again.");
33                 mainpage();
34             }
35         }
36         catch(Exception e)
37         {
38             System.out.println("Please enter an integer. Please try again.");
39             mainpage();
40         }
41     }
42 }
```

```
import java.util.Scanner;
import java.io.*;
class ResultPage
{
    static String dep;
    static String arr;
    static int pas;
    static String date;
    static Scanner sc= new Scanner(System.in);
    static BufferedReader brline = new BufferedReader(new
InputStreamReader(System.in));
    public static void enter(String dep,String arr,int pas,String date)
    {
        ResultPage.dep = dep;
        ResultPage.arr = arr;
        ResultPage.pas = pas;
        ResultPage.date = date;
        flight[0]=0;
        flight[1]=0;
        page1();
    }
    static  int flight[] = new int[2];
    static int dir;
    private static void page1()
    {

        try
        {
            java.util.Map<String,String> map = getmapofairport.main();
            System.out.println("\fDeparture Airport: "+map.get(dep)+"\n"+
                "Arrival Airport: "+map.get(arr)+"\n"+
                "Date: "+date+"\n"+
                "Passengers: "+pas);
            BufferedReader br = new BufferedReader(new
FileReader("flightlist1.txt"));
            int flag=0;
            String text;
            int i=1;
            int[] a = new int[3];
            while((text=br.readLine())!=null)
            {
                String text1 = text.substring(4,11);
                String text2 = dep.concat(" ".concat(arr));
                if(text1.equals((text2)))
                {
                    if(flag==0)
                    {
                        System.out.println("\nFlight No.    Departure Airport \t
Arrival Airport \t Departure Time \t Arrival Time \t    Cost(In Rupees) \t
Availability(No. of seats)");
```

```
47
System.out.println("-------------------------------------------------------------------
-------------------------------------------------------------------------------
--------------");
48                      flag=1;
49                  }
50                  System.out.println("   DF"+text.substring(0,3)+"\t
"+map.get(text.substring(4,7))+"\t\t    "+map.get(text.substring(8,11))+"\t\t     "+
51                      text.substring(12,18)+"\t\t
"+text.substring(18,24)+"\t\t"+text.substring(24)
+"\t\t\t"+checkseats(Integer.parseInt(text.substring(0,3)),date));
52                  a[i-1] = Integer.parseInt(text.substring(0,3));
53                  i++;
54              }
55          }
56          if(flag==0)
57          {
58              System.out.println("\nThere are no direct flights between the 2
locations you have chosen.\n"+
59                  "Please choose from one of the options below.");
60              dir=1;
61              flight = nodirect();
62
63              System.out.println("\nYou have chosen flights DF"+ flight[0]+" and
DF"+flight[1]+".");
64              if(checkseats(flight[0],date)>=pas &&
checkseats(flight[1],date)>=pas)
65              {
66                  System.out.println("There are seats available.");
67              }
68              else
69              {
70                  System.out.println("Seats are no available.");
71                  System.out.println("\nPlease choose one of the options:\n"+
72                      "1. To make a choice again.\n"+
73                      "2. To go back to the home page.");
74                  choicenoseats();
75                  return;
76              }
77          }
78          if(i>1)
79          {
80              System.out.println("\nPlease enter the Flight Number from the above
choices.\n");
81              flight[0] = enterfn(a);
82              dir=0;
83              System.out.println("\nYou have chosen the flight DF"+flight[0]);
84              if(checkseats(flight[0],date)>=pas)
85              {
86                  System.out.println("There are seats available.");
```

```
87                  }
88                  else
89                  {
90                      System.out.println("Seats are no available.");
91                      System.out.println("\nPlease choose one of the options:\n"+
92                          "1. To make a choice again.\n"+
93                          "2. To go back to the home page.");
94                      choicenoseats();
95                      return;
96                  }
97              }
98          }
99          catch(Exception e){}
100
101         System.out.println("\nPlease choose one of the options:\n"+
102             "1. To continue with the above flight(s) and enter the passengers'
    details.\n"+
103             "2. To make a choice again.\n"+
104             "3. To go back to the home page.");
105         choicemainpage();
106     }
107     static int c=0;
108
109     public static void choicenoseats()
110     {
111         System.out.print("Your choice: ");
112         try
113         {
114             c = Integer.parseInt(brline.readLine());
115         }
116         catch(Exception e)
117         {
118             System.out.println("Please enter an integer.");
119             choicemainpage();
120         }
121         switch(c)
122         {
123             case 1:
124             System.out.print("\f");
125             page1();
126             break;
127             case 2:
128             System.out.print("\f");
129             MainPage.mainpage();
130             break;
131             default:
132             System.out.println("Invalid choice. Please enter again.");
133             choicemainpage();
134         }
135     }
```

```
136
137     public static void choicemainpage()
138     {
139         System.out.print("Your choice: ");
140         try
141         {
142             c = Integer.parseInt(brline.readLine());
143         }
144         catch(Exception e)
145         {
146             System.out.println("Please enter an integer.");
147             choicemainpage();
148         }
149         switch(c)
150         {
151             case 1:
152             page2();
153             break;
154             case 2:
155             System.out.print("\f");
156             page1();
157             break;
158             case 3:
159             System.out.print("\f");
160             MainPage.mainpage();
161             break;
162             default:
163             System.out.println("Invalid choice. Please enter again.");
164             choicemainpage();
165         }
166     }
167     static String details[][];
168     public static void page2()
169     {
170         System.out.println("\fPlease enter the details of all the passengers: ");
171         details = Passengers.enterdetails(pas);
172         System.out.println("Are you sure you want to continue with the above
details? Once entered, these details cannot be changed!\n"+
173             "Please choose one of the following: \n"+
174             "1. To continue with the above details.\n"+
175             "2. To reset this page.\n"+
176             "3. To go back to the previous page.\n"+
177             "4. To go to the home page.");
178         choicepage2();
179     }
180
181     public static void choicepage2()
182     {
183         try
184         {
```

```
185             System.out.print("Your choice: ");
186             c = Integer.parseInt(brline.readLine());
187         }
188         catch(Exception e){
189             System.out.println("Please enter an integer.");
190             choicepage2();
191             return;
192         }
193         switch(c)
194         {
195             case 1:
196             page3();
197             break;
198             case 2:
199             page2();
200             break;
201             case 3:
202             page1();
203             break;
204             case 4:
205             MainPage.mainpage();
206             break;
207             default:
208             System.out.println("Invalid Choice. Please try again.");
209             choicepage2();
210             break;
211         }
212     }
213
214     public static void page3()
215     {
216         if(flight[1]==0)
217         {
218             System.out.println("\fPlease choose the seats for the passenger(s):\n"+
219                 "The seats which have a specific number on them are the ones which
    are available.\n"+
220                 "The seats which are marked with with 'XXX' have already been
    booked.\n"+
221                 "Please choose the seats which you want for each passenger.");
222             try
223             {
224                 int arr[][] = arrayofseats(flight[0]);
225                 for(int i=0;i<details.length;i++)
226                 {
227                     System.out.println("Passenger: "+details[i][0]);
228                     arraydisplay(flight[0]);
229                     System.out.println();
230                     String seat = seatnumber(flight[0]);
231                     arr = updatearray(seat,arr);
232                     details[i][2] = seat;
```

```
233                 System.out.println();
234             }
235             System.out.println("Please choose of the following: \n"+
236                 "1. To continue with the data given by you.\n"+
237                 "2. To reset this page.\n"+
238                 "3. To go back to the main menu.");
239             choicepage3(arr);
240         }
241         catch(Exception e)
242         {
243             System.out.println("An exception occured: "+ e);
244         }
245     }
246     else
247     {
248         System.out.println("\fPlease choose the seats for the passenger(s):\n"+
249             "The seats which have a specific number on them are the ones which
    are available.\n"+
250             "The seats which are marked with with 'XXX' have already been
    booked.\n"+
251             "Please choose the seats which you want for each passenger.");
252         try
253         {
254             System.out.println("\nFor Flight DF"+flight[0]);
255             int arr[][] = arrayofseats(flight[0]);
256             for(int i=0;i<details.length;i++)
257             {
258                 System.out.println("Passenger: "+details[i][0]);
259                 arraydisplay(flight[0]);
260                 System.out.println();
261                 String seat = seatnumber(flight[0]);
262                 arr = updatearray(seat,arr);
263                 details[i][2] = seat;
264                 System.out.println();
265             }
266             System.out.println("\nFor Flight DF"+flight[1]);
267             int[][] arr1 = arrayofseats(flight[1]);
268             for(int i=0;i<details.length;i++)
269             {
270                 System.out.println("Passenger: "+details[i][0]);
271                 arraydisplay(flight[1]);
272                 System.out.println();
273                 String seat = seatnumber(flight[1]);
274                 arr1 = updatearray(seat,arr1);
275                 details[i][2]+=" "+seat;
276                 System.out.println();
277             }
278             System.out.println("Please choose of the following: \n"+
279                 "1. To continue with the data given by you.\n"+
280                 "2. To reset this page.\n"+
```

```
                "3. To go back to the main menu.");
            choicepage3(arr,arr1);
        }
        catch(Exception e)
        {

        }

    }
}

public static void choicepage3(int[][] arr)
{
    try
    {
        System.out.print("Your choice: ");
        c = Integer.parseInt(brline.readLine());
    }
    catch(Exception e){
        System.out.println("Please enter an integer.");
        choicepage3(arr);
        return;
    }
    switch(c)
    {
        case 1:
        writearray(arr,flight[0]);
        page4();
        break;
        case 2:
        page3();
        break;
        case 3:
        MainPage.mainpage();
        break;
        default:
        System.out.println("Invalid Choice. Please try again.");
        choicepage3(arr);
        break;
    }
}

public static void choicepage3(int[][] arr,int[][] arr1)
{
    try
    {
        System.out.print("Your choice: ");
        c = Integer.parseInt(brline.readLine());
    }
    catch(Exception e){
```

```
331             System.out.println("Please enter an integer.");
332             choicepage3(arr);
333             return;
334         }
335         switch(c)
336         {
337             case 1:
338             writearray(arr,flight[0]);
339             writearray(arr1,flight[1]);
340             page4();
341             break;
342             case 2:
343             page3();
344             break;
345             case 3:
346             MainPage.mainpage();
347             break;
348             default:
349             System.out.println("Invalid Choice. Please try again.");
350             choicepage3(arr);
351             break;
352         }
353     }
354
355     public static void page4()
356     {
357         System.out.println("\fConfirmation Page\n"+
358             "Are you sure you want to confirm the booking?\n"+
359             "The details of the booking are:\n");
360         if(flight[1]==0)
361         {
362             java.util.Map<String,String> map = new
java.util.HashMap<String,String>();
363             try
364             {
365                 map = getmapofairport.main();
366             }
367             catch(Exception e){}
368             System.out.println("Date: "+date);
369             System.out.println("Source City: "+map.get(dep));
370             System.out.println("Arrival City: "+map.get(arr));
371             System.out.println("Is direct: True");
372             System.out.println("Flight Number: DF"+flight[0]);
373             String deptime="";
374             String arrtime="";
375             int cost=0;
376             try
377             {
378                 BufferedReader br = new BufferedReader(new
FileReader("flightlist1.txt"));
```

```
379                 String text;
380                 while((text = br.readLine())!=null)
381                 {
382                     if(Integer.parseInt(text.substring(0,3))==flight[0])
383                     {
384                         deptime = text.substring(12,17);
385                         arrtime = text.substring(18,23);
386                         cost = Integer.parseInt(text.substring(24));
387                         break;
388                     }
389                 }
390             }
391             catch(Exception e)
392             {}
393             System.out.println("\nDeparture Time: "+deptime);
394             System.out.println("Arrival Time: "+arrtime);
395             System.out.println("Total Cost: Rs"+(cost*details.length));
396             System.out.println("Number of passengers: "+details.length);
397             System.out.println("\nDetails of Passengers:");
398             System.out.println("\tName\t\tAge\tSeat No.\tMobile Number\tGender");
399             for(int i=0;i<details.length;i++)
400             {
401                 if(details[i][0].length()>14)
402 
System.out.println(details[i][0]+"\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+detail
s[i][3]+"\t"+details[i][4]);
403                 else
404 
System.out.println(details[i][0]+"\t\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+deta
ils[i][3]+"\t"+details[i][4]);
405                 }
406         }
407         else
408         {
409             java.util.Map<String,String> map = new
java.util.HashMap<String,String>();
410             try
411             {
412                 map = getmapofairport.main();
413             }
414             catch(Exception e){}
415             System.out.println("Date: "+date);
416             System.out.println("Source City: "+map.get(dep));
417             System.out.println("Arrival City: "+map.get(arr));
418             System.out.println("Is Direct: False");
419             System.out.println("\nFlight 1: DF"+flight[0]);
420             String dep1="";
421             String arr1="";
422             String deptime="";
423             String arrtime="";
```

```
424             int cost=0;
425             try
426             {
427                 BufferedReader br = new BufferedReader(new
    FileReader("flightlist1.txt"));
428                 String text;
429                 while((text = br.readLine())!=null)
430                 {
431                     if(Integer.parseInt(text.substring(0,3))==flight[0])
432                     {
433                         dep1 = map.get(text.substring(4,7));
434                         arr1 = map.get(text.substring(8,11));
435                         deptime = text.substring(12,17);
436                         arrtime = text.substring(18,23);
437                         cost = Integer.parseInt(text.substring(24));
438                         break;
439                     }
440                 }
441             }
442             catch(Exception e)
443             {}
444             System.out.println("Departure City: "+dep1);
445             System.out.println("Arrival City: "+arr1);
446             System.out.println("Departure Time: "+deptime);
447             System.out.println("Arrival Time: "+arrtime);
448             System.out.println();
449             System.out.println("Flight 2: DF"+flight[1]);
450             String dep2="";
451             String arr2="";
452             String arr1time="";
453             String dep1time="";
454             try
455             {
456                 BufferedReader br = new BufferedReader(new
    FileReader("flightlist1.txt"));
457                 String text;
458                 while((text = br.readLine())!=null)
459                 {
460                     if(Integer.parseInt(text.substring(0,3))==flight[1])
461                     {
462                         dep2 = map.get(text.substring(4,7));
463                         arr2 = map.get(text.substring(8,11));
464                         dep1time = text.substring(12,17);
465                         arr1time = text.substring(18,23);
466                         cost += Integer.parseInt(text.substring(24));
467                         break;
468                     }
469                 }
470             }
471             catch(Exception e){}
```

```
472             System.out.println("Departure City: "+dep2);
473             System.out.println("Arrival City: "+arr2);
474             System.out.println("Departure Time: "+dep1time);
475             System.out.println("Arrival Time: "+arr1time);
476             System.out.println();
477             System.out.println("Layover:");
478             System.out.println("City: "+dep2);
479             System.out.println("Time: "+
(Integer.parseInt(dep1time.substring(0,2))-Integer.parseInt(arrtime.substring(0,2)))
+" hours.");
480             System.out.println();
481             System.out.println("Total Cost: Rs"+(cost*details.length));
482             System.out.println();
483             System.out.println("Number of passengers: "+details.length);
484             System.out.println("\nDetails of Passengers:");
485             System.out.println("\tName\t\tAge\tSeat Nos.\tMobile Number\tGender");
486             for(int i=0;i<details.length;i++)
487             {
488                 if(details[i][0].length()>14)
489 
System.out.println(details[i][0]+"\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+detail
s[i][3]+"\t"+details[i][4]);
490                 else
491 
System.out.println(details[i][0]+"\t\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+deta
ils[i][3]+"\t"+details[i][4]);
492             }
493 
494         }
495         System.out.println("\nPlease choose one of the following:\n"+
496             "1.To confirm the booking.\n"+
497             "2.To go to the Main Page.");
498         choicepage4();
499     }
500 
501     public static void choicepage4()
502     {
503         try
504         {
505             System.out.print("Your choice: ");
506             c = Integer.parseInt(brline.readLine());
507         }
508         catch(Exception e){
509             System.out.println("Please enter an integer.");
510             choicepage2();
511             return;
512         }
513         switch(c)
514         {
515             case 1:
```

```
516             int code=writepassenger();
517             page5(code);
518             break;
519             case 2:
520             MainPage.mainpage();
521             break;
522             default:
523             System.out.println("Invalid Choice. Please try again.");
524             choicepage2();
525             break;
526         }
527     }
528
529     public static int writepassenger()
530     {
531         int code=0;
532         try
533         {
534             BufferedReader br = new BufferedReader(new
    FileReader("Passengers\\Code.txt"));
535             code = Integer.parseInt(br.readLine());
536             PrintWriter pw=new PrintWriter(new BufferedWriter(new
    FileWriter("Passengers\\Code.txt")));
537             pw.println(code+1);
538             pw.close();
539             br.close();
540         }
541         catch(Exception e){}
542         try
543         {
544             if(flight[1]==0)
545             {
546                 java.util.Map<String,String> map = new
    java.util.HashMap<String,String>();
547                 map = getmapofairport.main();
548                 PrintWriter pw=new PrintWriter(new BufferedWriter(new
    FileWriter("Passengers\\Pas"+code+".txt")));
549                 pw.println("Date: "+date);
550                 pw.println("Source City: "+map.get(dep));
551                 pw.println("Arrival City: "+map.get(arr));
552                 pw.println("Is direct: True");
553                 pw.println("Flight Number: DF"+flight[0]);
554                 String deptime="";
555                 String arrtime="";
556                 int cost=0;
557                 try
558                 {
559                     BufferedReader br = new BufferedReader(new
    FileReader("flightlist1.txt"));
560                     String text;
```

```
561                     while((text = br.readLine())!=null)
562                     {
563                         if(Integer.parseInt(text.substring(0,3))==flight[0])
564                         {
565                             deptime = text.substring(12,17);
566                             arrtime = text.substring(18,23);
567                             cost = Integer.parseInt(text.substring(24));
568                             break;
569                         }
570                     }
571                 }
572                 catch(Exception e)
573                 {}
574                 pw.println("\nDeparture Time: "+deptime);
575                 pw.println("Arrival Time: "+arrtime);
576                 pw.println("Total Cost: Rs"+(cost*details.length));
577                 pw.println("Number of passengers: "+details.length);
578                 pw.println("\nDetails of Passengers:");
579                 pw.println("\tName\t\tAge\tSeat No.\tMobile Number\tGender");
580                 for(int i=0;i<details.length;i++)
581                 {
582                     if(details[i][0].length()>14)
583 
pw.println(details[i][0]+"\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+details[i][3]+
"\t"+details[i][4]);
584                     else
585 
pw.println(details[i][0]+"\t\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+details[i][3
]+"\t"+details[i][4]);
586                 }
587                 pw.close();
588             }
589             else
590             {
591                 PrintWriter pw=new PrintWriter(new BufferedWriter(new
FileWriter("Passengers\\Pas"+code+".txt")));
592                 java.util.Map<String,String> map = new
java.util.HashMap<String,String>();
593                 try
594                 {
595                     map = getmapofairport.main();
596                 }
597                 catch(Exception e){}
598                 pw.println("Date: "+date);
599                 pw.println("Source City: "+map.get(dep));
600                 pw.println("Arrival City: "+map.get(arr));
601                 pw.println("Is Direct: False");
602                 pw.println("\nFlight 1: DF"+flight[0]);
603                 String dep1="";
604                 String arr1="";
```

```
605                 String deptime="";
606                 String arrtime="";
607                 int cost=0;
608                 try
609                 {
610                     BufferedReader br = new BufferedReader(new
    FileReader("flightlist1.txt"));
611                     String text;
612                     while((text = br.readLine())!=null)
613                     {
614                         if(Integer.parseInt(text.substring(0,3))==flight[0])
615                         {
616                             dep1 = map.get(text.substring(4,7));
617                             arr1 = map.get(text.substring(8,11));
618                             deptime = text.substring(12,17);
619                             arrtime = text.substring(18,23);
620                             cost = Integer.parseInt(text.substring(24));
621                             break;
622                         }
623                     }
624                 }
625                 catch(Exception e)
626                 {}
627                 pw.println("Departure City: "+dep1);
628                 pw.println("Arrival City: "+arr1);
629                 pw.println("Departure Time: "+deptime);
630                 pw.println("Arrival Time: "+arrtime);
631                 pw.println();
632                 pw.println("Flight 2: DF"+flight[1]);
633                 String dep2="";
634                 String arr2="";
635                 String arr1time="";
636                 String dep1time="";
637                 try
638                 {
639                     BufferedReader br = new BufferedReader(new
    FileReader("flightlist1.txt"));
640                     String text;
641                     while((text = br.readLine())!=null)
642                     {
643                         if(Integer.parseInt(text.substring(0,3))==flight[1])
644                         {
645                             dep2 = map.get(text.substring(4,7));
646                             arr2 = map.get(text.substring(8,11));
647                             dep1time = text.substring(12,17);
648                             arr1time = text.substring(18,23);
649                             cost += Integer.parseInt(text.substring(24));
650                             break;
651                         }
652                     }
```

```
653                 }
654                 catch(Exception e){}
655                 pw.println("Departure City: "+dep2);
656                 pw.println("Arrival City: "+arr2);
657                 pw.println("Departure Time: "+dep1time);
658                 pw.println("Arrival Time: "+arr1time);
659                 pw.println();
660                 pw.println("Layover:");
661                 pw.println("City: "+dep2);
662                 pw.println("Time: "+
(Integer.parseInt(dep1time.substring(0,2))-Integer.parseInt(arrtime.substring(0,2)))
+" hours.");
663                 pw.println();
664                 pw.println("Total Cost: Rs"+(cost*details.length));
665                 pw.println();
666                 pw.println("Number of passengers: "+details.length);
667                 pw.println("\nDetails of Passengers:");
668                 pw.println("\tName\t\tAge\tSeat Nos.\tMobile Number\tGender");
669                 for(int i=0;i<details.length;i++)
670                 {
671                     if(details[i][0].length()>14)
672 
pw.println(details[i][0]+"\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+details[i][3]+
"\t"+details[i][4]);
673                     else
674 
pw.println(details[i][0]+"\t\t"+details[i][1]+"\t"+details[i][2]+"\t\t"+details[i][3
]+"\t"+details[i][4]);
675                 }
676                 pw.close();
677             }
678         }
679         catch(Exception e)
680         {}
681 
682         return code;
683     }
684 
685     public static void page5(int code)
686     {
687         try
688         {
689         System.out.println("\fCongratulations.\n"+
690                            "Your booking has been confirmed!\n"+
691                            "Your passenger code is "+code+"\n"+
692                            "Please remember the passenger code. It contains all
details of the flight.\n"+
693                            "You can access the details from the 'Previous Booking'
using the code.\n\n"+
694                            "Payment details : \n"+
```

```
695                             "Please send the money specified before in one of the 2
following ways: \n"+
696                             "1. Please send the money to one of the airports which
we serve.\n"+
697                             "   The money can be deposited in our counter and along
with your passenger number.\n"+
698                             "2. Please transfer the money to our bank account.\n"+
699                             "   Our bank details are:\n"+
700                             "     Account Number: 3487210563\n"+
701                             "     Account Name: Domestic AirServices Pvt. Ltd.\n"+
702                             "     IFSC code: HSBC0400002\n"+
703                             "     Account Type: Current Account.\n"+
704                             "   Please transfer the money to this account with your
passenger code in the comments section.\n\n"+
705                             "Faliure to send the money 3 days from now will result
in the termination of the booking.\n"+
706                             "For any queries, please dont hesitate to contact us
at:\n"+
707                             "  Email: inquiry@domesticairservices.in\n"+
708                             "  Phone: +91 9830376021\n\n"+
709                             "Please press any character and enter to go back to main
menu.");
710         brline.readLine();
711         MainPage.mainpage();
712         }
713         catch(Exception e){}
714     }
715
716     public static int[][] updatearray(String seat, int[][] arr)
717     {
718         int num=0;
719         int pos=0;
720         if(seat.length()==2)
721         {
722             num = Integer.parseInt(seat.charAt(0)+"");
723             char c= seat.charAt(1);
724             if(c=='A')
725             {
726                 pos=1;
727             }
728             else if(c=='B')
729             {
730                 pos=2;
731             }
732             else if(c=='C')
733             {
734                 pos= 3;
735             }
736             else if(c=='D')
737             {
```

```
738                 pos= 4;
739             }
740             else if(c=='E')
741             {
742                 pos= 5;
743             }
744             else if(c=='F')
745             {
746                 pos= 6;
747             }
748         }
749         else
750         {
751             num =
(Integer.parseInt(seat.charAt(0)+"")*10)+Integer.parseInt(seat.charAt(1)+"");
752             char c= seat.charAt(2);
753             if(c=='A')
754             {
755                 pos=1;
756             }
757             else if(c=='B')
758             {
759                 pos=2;
760             }
761             else if(c=='C')
762             {
763                 pos= 3;
764             }
765             else if(c=='D')
766             {
767                 pos= 4;
768             }
769             else if(c=='E')
770             {
771                 pos= 5;
772             }
773             else if(c=='F')
774             {
775                 pos= 6;
776             }
777         }
778         arr[pos-1][num-1]=1;
779         return arr;
780     }
781
782     public static String seatnumber(int f)
783     {
784         try
785         {
786             System.out.print("Please enter the seat number for this passenger: ");
```

```
787             String str = brline.readLine();
788             str = str.trim();
789             str = str.toUpperCase();
790             String arr[] = arrayofavailableseats(f);
791             for(int i=0;i<arr.length;i++)
792             {
793                 if(arr[i]!=null&&str.equalsIgnoreCase(arr[i]))
794                     return str;
795             }
796             System.out.println("Invalid seat number. Please enter again.");
797             return seatnumber(f);
798         }
799         catch(Exception e)
800         {
801             System.out.println(e);
802         }
803         return null;
804     }
805
806     public static int[][] arrayofseats(int f) throws IOException
807     {
808         BufferedReader br = new BufferedReader(new
    FileReader("Data\\seats"+date+".txt"));
809         String text;
810         int i=1;
811         int j=0;
812         int arr[][] = new int[6][30];
813         while((text=br.readLine())!=null)
814         {
815             if(i == (f-101)*8 +3)
816             {
817                 for(int k=1;k<=6;k++)
818                 {
819                     for(int l=0;l<30;l++)
820                     {
821                         arr[j][l] = Integer.parseInt(text.charAt(l*2)+"");
822                     }
823                     j++;
824                     text = br.readLine();
825                 }
826                 break;
827             }
828             i++;
829         }
830         return arr;
831     }
832
833     public static void arraydisplay(int f)
834     {
835         String str[] = {"A","B","C","D","E","F"};
```

```
836         int arr[][] = new int[6][30];
837
838         try{
839             arr= arrayofseats(f);
840         }
841         catch(Exception e)
842         {}
843         for(int i=1;i<=119;i++)
844         {
845             System.out.print("-");
846         }
847         System.out.println();
848         for(int i=0;i<6;i++)
849         {
850             for(int j=29;j>=0;j--)
851             {
852                 if(arr[i][j]==0)
853                 {
854                     if((j+1)<10)
855                         System.out.print("0"+(j+1)+str[i]+" ");
856                     else
857                         System.out.print((j+1)+str[i]+" ");
858                 }
859                 else
860                     System.out.print("XXX ");
861             }
862             if(i<=2)
863             {
864                 for(int j =0;j<i;j++)
865                 {
866                     System.out.print(" ");
867                 }
868                 System.out.print("\\\\");
869             }
870             else
871             {
872                 for(int j =5;j>i;j--)
873                 {
874                     System.out.print(" ");
875                 }
876                 System.out.print("//");
877             }
878             System.out.println();
879             if(i==2)
880             {
881
882                 System.out.println();
883             }
884         }
885         for(int i=1;i<=119;i++)
```

```
886             {
887                 System.out.print("-");
888             }
889         }
890 
891         public static String[] arrayofavailableseats(int f)
892         {
893             String[] arr = new String[234];
894             int a[][] = new int[6][30];
895             try{
896                 a= arrayofseats(f);
897             }
898             catch(Exception e)
899             {}
900             String str[] = {"A","B","C","D","E","F"};
901             int c=0;
902             for(int i=0;i<6;i++)
903             {
904                 for(int j=0;j<30;j++)
905                 {
906                     if(a[i][j]==0)
907                     {
908                         if((j+1)<10)
909                         {
910                             arr[c]="0"+ (j+1) + str[i];
911                             c++;
912                             arr[c] = (j+1)+str[i];
913                         }
914                         else
915                             arr[c]=(j+1) + str[i];
916                         c++;
917                     }
918                 }
919             }
920             return arr;
921         }
922 
923         public static int[] nodirect()
924         {
925             int count=0;
926             int counted[][] = new int[10][2];
927             String arr[] = {"DEL","BOM","BLR","HYD","MAA"};
928             System.out.println("Option \t Flight No.    Departure Airport \t Arrival
    Airport \t Departure Time \t Arrival Time \t    Cost(In Rupees) \t Availability(No.
    of seats)");
929             for(int i=0;i<5;i++)
930             {
931                 try
932                 {
933                     java.util.Map<String,String> map = getmapofairport.main();
```

```
934                 BufferedReader br = new BufferedReader(new
    FileReader("flightlist1.txt"));
935                 String text;
936
937                 while((text=br.readLine())!=null)
938                 {
939                     String text1 = text.substring(4,11);
940                     String text2 = dep.concat(" ".concat(arr[i]));
941                     if(text1.equalsIgnoreCase(text2))
942                     {
943                         int arrtime=Integer.parseInt(text.substring(18,20));
944                         BufferedReader br1 = new BufferedReader(new
    FileReader("flightlist1.txt"));
945                         String text3;
946                         while((text3=br1.readLine())!=null)
947                         {
948                             text1 = text3.substring(4,11);
949                             text2 = arr[i].concat(" ".concat(ResultPage.arr));
950                             if(text1.equalsIgnoreCase(text2))
951                             {
952                                 int arrtime1 =
    Integer.parseInt(text3.substring(12,14));
953                                 if(arrtime1-arrtime>0)
954                                 {
955                                     counted[count][0] =
    Integer.parseInt(text.substring(0,3));
956                                     counted[count][1] =
    Integer.parseInt(text3.substring(0,3));
957                                     count = count+1;
958                                     System.out.println("  "+ count + "
    DF"+text.substring(0,3)+"\t "+map.get(text.substring(4,7))+"\t\t
    "+map.get(text.substring(8,11))+"\t\t     "+
959                                         text.substring(12,18)+"\t\t
    "+text.substring(18,24)+"\t\t"+text.substring(24)
    +"\t\t\t"+checkseats(Integer.parseInt(text.substring(0,3)),date));
960                                     System.out.println("
    DF"+text3.substring(0,3)+"\t "+map.get(text3.substring(4,7))+"\t\t
    "+map.get(text3.substring(8,11))+"\t\t     "+
961                                         text3.substring(12,18)+"\t\t
    "+text3.substring(18,24)+"\t\t"+text3.substring(24)
    +"\t\t\t"+checkseats(Integer.parseInt(text3.substring(0,3)),date));
962                                     System.out.println();
963                                 }
964                             }
965                         }
966                     }
967                 }
968             }
969             catch(Exception e)
970             {}
```

```
971             }
972             System.out.println("Please choose the option of the sequence of flights
    which you want to take: ");
973             return nodirectchoice(counted, count);
974         }
975
976         public static int[] nodirectchoice(int[][] arr, int i)
977         {
978             try
979             {
980                 System.out.print("Your choice: ");
981                 c = Integer.parseInt(brline.readLine());
982             }
983             catch(Exception e)
984             {
985                 System.out.println("Please enter an integer.");
986                 nodirectchoice(arr,i);
987             }
988             if(c<1 || c>i)
989             {
990                 System.out.println("Please enter a valid choice.");
991                 return nodirectchoice(arr,i);
992             }
993             return arr[c-1];
994         }
995
996         public static int enterfn(int[] arr)
997         {
998             System.out.print("The preferred flight No. is: DF");
999             int i = -1;
1000            try
1001            {
1002                i = sc.nextInt();
1003            }
1004            catch(Exception e)
1005            {
1006                System.out.println("Wrong choice. Please enter an Integer.");
1007                return enterfn(arr);
1008            }
1009
1010            for(int j=0;j<arr.length;j++)
1011            {
1012                if(arr[j]==i)
1013                    return i;
1014            }
1015            System.out.println("\nWrong choice. Please try again. ");
1016            return enterfn(arr);
1017        }
1018
1019        public static int checkseats(int f, String date) throws IOException
```

```
1020    {
1021        BufferedReader br = new BufferedReader(new
    FileReader("Data\\seats"+date+".txt"));
1022        String text;
1023        int i=1;
1024        while((text=br.readLine())!=null)
1025        {
1026            if(i == (f-101)*8 +2)
1027            {
1028                break;
1029            }
1030            i++;
1031        }
1032        return Integer.parseInt(text);
1033    }
1034
1035    public static void writearray(int[][] arr, int f)
1036    {
1037        try
1038        {
1039            BufferedReader br = new BufferedReader(new
    FileReader("Data\\seats"+date+".txt"));
1040            PrintWriter pw=new PrintWriter(new BufferedWriter(new
    FileWriter("temp.txt")));
1041            int i=1;String text;
1042            while((text=br.readLine())!=null)
1043            {
1044                if(i == (f-101)*8 +2)
1045                {
1046                    int s=Integer.parseInt(text);
1047                    for(int j=1;j<=pas;j++)
1048                    s--;
1049                    pw.println(s);
1050                    text = br.readLine();
1051                    for(int k=0;k<5;k++)
1052                    {
1053                        String str="";
1054                        for(int l=0;l<29;l++)
1055                        {
1056                            str+=arr[k][l]+" ";
1057                        }
1058                        str+=arr[k][29];
1059                        pw.println(str);
1060                        text = br.readLine();
1061                    }
1062                    {
1063                        String str="";
1064                        for(int l=0;l<29;l++)
1065                        {
1066                            str+=arr[5][l]+" ";
```

```
                    }
                    str+=arr[5][29];
                    pw.println(str);
                }
                i++;
            }
            else
            {
                i++;
                pw.println(text);
            }
        }
        br.close();
        pw.close();
        BufferedReader br1 = new BufferedReader(new FileReader("temp.txt"));
        PrintWriter pw1=new PrintWriter(new BufferedWriter(new
FileWriter("Data\\seats"+date+".txt")));
        while((text=br1.readLine())!=null)
        {
            pw1.println(text);
        }
        br1.close();
        pw1.close();
    }
    catch(Exception e)
    {}
}

public static void writetofile()
{
    int arr[][] = new int[6][30];
    arr[2][15] = 1;
    arr[3][12] = 1;
    writearray(arr,101);
}
}
```

```
1  import java.io.*;
2  class Passengers
3  {
4      static BufferedReader brline = new BufferedReader(new
   InputStreamReader(System.in));
5      //String mobile;
6      public static String[][] enterdetails(int pas)
7      {
8          System.out.println();
9          String details[][] = new String[pas][5];
10         for(int i=1;i<=pas;i++)
11         {
12             System.out.println("Passenger#"+i);
13             details[i-1][0] = name();
14             details[i-1][1] = ""+getage();
15             details[i-1][3] = mobno();
16             details[i-1][4] = gender();
17             System.out.println();
18         }
19         return details;
20     }
21
22     public static String name()
23     {
24         try{
25             System.out.print("Enter Name:");
26             String FName=brline.readLine().trim();
27             FName=" "+FName;
28             int flag=0;
29             String str="";
30             for(int i=0;i<FName.length();i++)
31             {
32                 if(FName.charAt(i)==' ')
33                 {
34                     flag=1;
35                     str+=Character.toUpperCase(FName.charAt(i));
36                 }
37                 else if(flag==1)
38                 {
39                     str+=FName.charAt(i);
40                     flag=0;
41                 }
42                 else
43                 {
44                     str+=Character.toLowerCase(FName.charAt(i));
45                 }
46                 }
47             return str.trim();
48         }
49         catch(Exception e){}
```

```
50          return name();
51      }
52
53      public static String gender()
54      {
55          char gen=' ';
56          System.out.println("Enter Gender:");
57          System.out.println("Enter M for Male");
58          System.out.println("Enter F for Female");
59          System.out.print("Your Gender: ");
60          try{
61              gen=brline.readLine().toUpperCase().charAt(0);
62          }
63          catch(Exception e){}
64          if(gen!='M' && gen!='F')
65          {
66              System.out.println("Please enter valid gender");
67              return gender();
68          }
69          String gen1="";
70          if (gen=='M')
71          {
72              gen1="Male";
73          }
74          else if(gen=='F')
75          {
76              gen1="Female";
77          }
78          return gen1;
79      }
80
81      static int age;
82      public static int getage()
83      {
84          System.out.print("Please enter your age: ");
85          try
86          {
87              age = Integer.parseInt(brline.readLine());
88              return age;
89          }
90          catch(Exception e)
91          {
92              System.out.println("Please enter an integer.");
93              return getage();
94          }
95      }
96
97      static String mobileno;
98      public static String mobno()
99      {
```

```
100         System.out.print("Enter Mobile Number without country code or spaces:");
101
102         try
103         {
104             mobileno = brline.readLine();
105         }
106         catch(Exception e)
107         {
108             return mobno();
109         }
110         int l= mobileno.length();
111         if(l!=10)
112         {
113             System.out.println("Please Enter Valid Mobile Number");
114             return mobno();
115         }
116         char c;
117         for(int i=0;i<l;i++)
118         {
119             c=mobileno.charAt(i);
120             if(Character.isLetter(c))
121             {
122                 System.out.println("Please enter valid phone number");
123                 return mobno();
124             }
125
126         }
127         return mobileno;
128     }
129 }
```

VARIABLE DESCRIPTION TABLES

CLASS: Main

Name	Scope	Type	Description
mp	main()	Java Object	Object to call the main function of the class MainPage

CLASS: MainPage

Name	Scope	Type	Description
c	class variable	int	Stores the choice of the user in the menu driven parts
brline	class variable	BufferedReader	Used to accept input from the user

CLASS: FlightBooking

Name	Scope	Type	Description
dep	class variable	String	Stores the departure airport
arr	class variable	String	Stores the arrival airport
pas	class variable	int	Stores the number of passengers
date	class variable	String	Stores the date of travel
d	enterdate()	int	Stores the date(dd)
m	enterdate()	int	Stores the month(mm)
y	enterdate()	int	Stores the year(yyyy)
dc	checkdate()	int	Stores the date of the day of use(dd)
mc	checkdate()	int	Stores the month of the day of use(mm)
yc	checkdate()	int	Stores the year of the day of use(dd)
mm	checkdate()	int	Used to check if the day can be accepted or not
c	class variable	int	Used to accept a choice for the menu driven parts
p	passengers()	int	Used to accept the number of passengers and check if correct or not(less than 10 is correct)
d	enterdate()	int	Used to accept the date(dd)
m	enterdate()	int	Used to accept the month(mm)
y	enterdate()	int	Used to accept the year(yyyy)
str	enterdate()	String	Used to store the date as entered by the user

CLASS: PreviousBooking

Name	Scope	Type	Description
n	class variable	int	Accepts the Passenger code from the user
br	mainpage()	BufferedReader Object	Used to iterate through all the lines of the file: (“Pas”+n+”.txt”)

CLASS: OurNetwork

Name	Scope	Type	Description
c	class variable	int	Used to accept choices in the menu driven part
br	schedule()	BufferedReader Object	Used to iterate through lines of the file: “flightlist1.txt”
br	dirconnected()	BufferedReader Object	Used to iterate through lines of the file: “DirectFlights.txt”

CLASS: About Us

Name	Scope	Type	Description
sc	mainpage()	Scanner class	Object to accept input from the user

CLASS: getmapofairport

Name	Scope	Type	Description
map	main()	java.util.Map object(<String,String>)	Stores all the IATA codes of airports with respect to their names
br	main()	BufferedReader Object	Iterates through all the lines of the file: “airportscode.txt”
bw	main()	BufferedReader Object	Iterates through all the lines of the file: “airportsname.txt”

CLASS: ResultPage

Name	Scope	Type	Description
dep	class variable	String	Stores the departure airport
arr	class variable	String	Stores the arrival airport
pas	class variable	int	Stores the number of passengers
date	class variable	String	Stores the date of travel
flight	class variable	int[]	Stores the flight(s) which the user chooses
dir	class variable	int	Stores whether the journey has a direct or indirect flight
flag	page1()	int	Stores whether there is a direct flight or not
text	page1()	String	Stores each line of the file “flightlist1.txt”
i	page1()	int	while loop counter variable
a	page1()	int[]	Stores the options of the flights
text1	page1()	String	Stores the departure and arrival airports in the same form as in the file
text2	page1()	String	Stores the departure and arrival airports in the same form as in the file as entered by the user
c	class variable	int	Stores the choice of the user in the menu driven part
arr	page3()	int[][]	Stores the array of seats from the file
i	page3()	int	For loop counter variable
arr1	page3()	int[][]	Stores the array of seats from the file
arr	choicepage3()	int[][]	Stores the array of the seats
cost	page4()	int	Stores the total cost of the flights for the passengers
code	choicepage4()	int	Stores the passenger code of the booking
code	writepassenger()	int	Stores the passenger code of the booking
cost	writepassenger()	int	Stores the total cost of the flights for the passengers
i	writepassenger()	int	For loop counter variable
code	page5()	int	Stores the passenger code of the booking
seat	updatearray()	String	Stores the seat number chosen by the user
arr	updatearray()	int[][]	Stores the array of seats
num	updatearray()	int	Stores the alphabetical part of the seat number
pos	updatearray()	int	Stores the numeral part of the seat number
c	updatearray()	char	Stores the characters of the entered seat number
f	seatnumber()	int	Stores the flight number
f	arrayofseats()	int	Stores the flight number
i	arrayofseats()	int	Loop counter variable

j	arrayofseats()	int	Loop counter variable
arr	arrayofseats()	int[][]	Stores the array of seats
k	arrayofseats()	int	For loop counter variable
l	arrayofseats()	int	For loop counter variable
f	arraydisplay()	int	Stores the flight number
arr	arraydisplay()	int[][]	Stores the array of seats
i	arraydisplay()	int	For loop counter variable
j	arraydisplay()	int	For loop counter variable
f	arrayofavailableseats()	int	Stores the flight number
a	arrayofavailableseats()	int[][]	Stores the array of seats
c	arrayofavailableseats()	int	Stores the choice of the user
i	arrayofavailableseats()	int	For loop counter variable
j	arrayofavailableseats()	int	For loop counter variable
count	nodirect()	int	Counter variable
counted	nodirect()	int[][]	Stores the sequence of flights for non-direct routes
arr	nodirect()	String[]	Stores the airports which offers layovers
i	nodirect()	int	For loop counter variable
text	nodirect()	String	Stores each line of the file “flightlist1.txt”
text1	nodirect()	String	Stores the departure and arrival airports in the same form as in the file
text2	nodirect()	String	Stores the departure and arrival airports in the same form as in the file as entered by the user
arrtime	nodirect()	int	Stores the arrival time of the first flight
arrtime1	nodirect()	int	Stores the departure time of the second flight
arr	nodirectchoice()	int[][]	Stores the sequence of flights for non-direct routes
i	nodirectchoice()	int	Stores the number of options in arr
arr	enterfn()	int	Stores the number of the options for flights
i	enterfn()	int	Stores the flight number as entered by the user
j	enterfn()	int	For loop counter variable
f	checkseats()	int	Stores the flight number
date	checkseats()	String	Stores the date
i	checkseats()	int	Counter variable
arr	writearray()	int[][]	Stores the array of seats
f	writearray()	int	Stores the flight number
i	writearray()	int	Counter variable
s	writearray()	int	Stores the number of seats in the flight
k	writearray()	int	For loop counter variable
l	writearray()	int	For loop counter variable
arr	writetofile()	int[][]	Stores the array of seats

CLASS: Passengers

Name	Scope	Type	Description
l	enterdetails()	int	For loop counter variable
details	enterdetails()	String[][]	Stores the details of all the passengers
Fname	name()	String	Stores the name of the passenger
flag	name()	int	Temporary variable to check for spaces
str	name()	String	Stores the new name in Sentence Case
i	name()	int	For loop counter variable
gen	gender()	char	Stores the choice of gender of the passenger
age	class variable	int	Stores the age of the passenger
mobileno	class variable	String	Stores the mobile number of the passenger
l	mobno()	int	Stores the length of mobileno
c	mobno()	char	Stores each character of mobileno to check if all the characters are digits or not
i	mobno()	int	For loop counter variable

Pre-written files used.

The following files and their content had been prewritten for the use of the program:

- flightlist1: This file contains the entire list of flights, its schedule and the price. The contents are:

  101 DEL BOM 05:00 07:00 4000
  102 DEL BOM 12:00 14:00 4500
  103 DEL BOM 18:00 20:00 4500
  104 BOM DEL 08:00 10:00 5000
  105 BOM DEL 15:00 17:00 5000
  106 BOM DEL 21:00 23:00 4500
  107 DEL CCU 06:00 08:00 6000
  108 DEL CCU 13:00 15:00 6500
  109 DEL CCU 19:00 21:00 7000
  110 CCU DEL 09:00 11:00 8000
  111 CCU DEL 16:00 18:00 8500
  112 CCU DEL 21:00 23:00 8000
  113 BOM CCU 05:00 08:00 9000
  114 BOM CCU 11:00 14:00 9500
  115 BOM CCU 20:00 23:00 9500
  116 CCU BOM 08:00 11:00 10000
  117 CCU BOM 14:00 17:00 10000
  118 CCU BOM 17:00 20:00 10000
  119 DEL MAA 07:00 10:00 10000
  120 DEL MAA 13:00 16:00 11000
  121 DEL MAA 18:00 21:00 11000
  122 MAA DEL 10:00 13:00 13000
  123 MAA DEL 15:00 18:00 13000
  124 MAA DEL 20:00 23:00 13000
  125 BOM MAA 09:00 11:00 8000
  126 BOM MAA 15:00 17:00 8800
  127 BOM MAA 19:00 21:00 8800
  128 MAA BOM 05:00 07:00 6000
  129 MAA BOM 11:00 13:00 9000
  130 MAA BOM 21:00 23:00 6000
  131 CCU MAA 09:00 12:00 12000
  132 CCU MAA 17:00 20:00 12000
  133 MAA CCU 13:00 14:00 11000
  134 MAA CCU 20:00 23:00 10000

135 DEL COK 07:00 10:00 15000
136 COK DEL 13:00 16:00 16000
137 BOM COK 11:00 13:00 7000
138 COK BOM 16:00 18:00 8000
139 MAA COK 07:00 08:00 6000
140 MAA COK 13:00 14:00 7000
141 COK MAA 10:00 11:00 7000
142 COK MAA 15:00 16:00 7000
143 DEL GOI 09:00 11:00 8000
144 GOI DEL 12:00 14:00 7500
145 BOM GOI 09:00 10:00 5000
146 BOM GOI 14:00 15:00 6000
147 GOI BOM 11:00 12:00 5500
148 GOI BOM 18:00 19:00 6500
149 MAA GOI 13:00 14:00 8000
150 GOI MAA 15:00 16:00 8000
151 DEL BLR 10:00 12:00 7000
152 DEL BLR 15:00 17:00 7000
153 DEL BLR 19:00 21:00 7000
154 BLR DEL 13:00 15:00 6500
155 BLR DEL 17:00 19:00 6600
156 BLR DEL 05:00 07:00 6000
157 BOM BLR 06:00 07:00 4500
158 BOM BLR 13:00 14:00 5000
159 BOM BLR 19:00 20:00 5000
160 BLR BOM 09:00 10:00 5000
161 BLR BOM 16:00 17:00 6000
162 BLR BOM 22:00 23:00 6000
163 CCU BLR 12:00 14:00 8000
164 CCU BLR 17:00 19:00 8000
165 BLR CCU 08:00 10:00 7500
166 BLR CCU 13:00 15:00 7500
167 MAA BLR 09:00 10:00 6000
168 MAA BLR 13:00 14:00 7500
169 MAA BLR 17:00 18:00 6500
170 BLR MAA 05:00 06:00 5500
171 BLR MAA 11:00 12:00 6000
172 BLR MAA 15:00 16:00 6500
173 COK BLR 06:00 08:00 5000
174 COK BLR 13:00 15:00 5500
175 BLR COK 09:00 11:00 5500
176 BLR COK 16:00 18:00 5500
177 GOI BLR 10:00 11:00 5000
178 GOI BLR 14:00 15:00 6000

179 BLR GOI 12:00 13:00 5500
180 BLR GOI 16:00 17:00 5500
181 DEL HYD 06:00 08:00 9000
182 DEL HYD 14:00 16:00 9000
183 DEL HYD 20:00 22:00 9000
184 HYD DEL 10:00 12:00 8500
185 HYD DEL 17:00 19:00 8500
186 HYD DEL 21:00 23:00 7500
187 BOM HYD 09:00 10:00 7000
188 BOM HYD 14:00 15:00 7500
189 HYD BOM 12:00 13:00 8000
190 HYD BOM 17:00 18:00 8000
191 CCU HYD 10:00 12:00 10000
192 CCU HYD 15:00 17:00 11000
193 HYD CCU 13:00 15:00 10500
194 HYD CCU 18:00 20:00 11000
195 MAA HYD 07:00 09:00 6000
196 MAA HYD 11:00 12:00 6500
197 HYD MAA 09:00 11:00 5500
198 HYD MAA 13:00 15:00 6000
199 GOI HYD 16:00 17:00 7000
200 HYD GOI 18:00 19:00 8000
201 COK HYD 07:00 08:00 7500
202 HYD COK 11:00 12:00 8000
203 BLR HYD 05:00 06:00 7000
204 BLR HYD 10:00 11:00 7500
205 BLR HYD 14:00 15:00 7000
206 HYD BLR 07:00 08:00 6500
207 HYD BLR 12:00 13:00 7500
208 HYD BLR 16:00 17:00 7500
209 DEL ATQ 18:00 19:00 6000
210 DEL ATQ 09:00 10:00 7000
211 ATQ DEL 07:00 08:00 6500
212 ATQ DEL 11:00 12:00 6500
213 BOM ATQ 20:00 22:00 7000
214 ATQ BOM 07:00 09:00 7500
215 MAA TRV 08:00 09:00 6000
216 MAA TRV 13:00 14:00 7000
217 TRV MAA 11:00 12:00 6500
218 TRV MAA 15:00 17:00 7000
219 COK TRV 08:00 10:00 5000
220 COK TRV 12:00 13:00 6000
221 TRV COK 10:00 11:00 5500
222 TRV COK 14:00 15:00 5500

223 BLR TRV 16:00 17:00 6000
224 TRV BLR 19:00 20:00 7000
225 HYD TRV 08:00 10:00 8000
226 TRV HYD 12:00 14:00 8500
227 BOM TRV 10:00 12:00 9000
228 TRV BOM 16:00 18:00 9500

- airportscode: Contains the list of IATA codes of the 10 airports:-

  DEL
  BOM
  CCU
  MAA
  COK
  GOI
  BLR
  HYD
  ATQ
  TRV

- airportsname: Contains the list of names of airports corresponding to their IATA codes in airportscode:-

  New Delhi
  Mumbai
  Kolkata
  Chennai
  Cochin
  Goa-Panaji
  Bangalore
  Hyderabad
  Amritsar
  Trivandrum

- Temp: this is a temporary file used for the replacement of data in the auto-generated files.

- DirectFlights: Contains all the direct flights operated by the airline with the frequency per day:-

  DEL BOM 3
  BOM DEL 3
  DEL CCU 3
  CCU DEL 3
  BOM CCU 3
  CCU BOM 3
  DEL MAA 3
  MAA DEL 3
  BOM MAA 3
  MAA BOM 3
  CCU MAA 2
  MAA CCU 2
  DEL COK 1
  COK DEL 1
  BOM COK 1
  COK BOM 1
  MAA COK 2
  COK MAA 2
  DEL GOI 1
  GOI DEL 1
  BOM GOI 2
  GOI BOM 2
  MAA GOI 1
  GOI MAA 1
  DEL BLR 3
  BLR DEL 3
  BOM BLR 3
  BLR BOM 3
  CCU BLR 2
  BLR CCU 2
  MAA BLR 3
  BLR MAA 3
  COK BLR 2
  BLR COK 2
  GOI BLR 2
  BLR GOI 2
  DEL HYD 3
  HYD DEL 3
  BOM HYD 2
  HYD BOM 2

CCU HYD 2
HYD CCU 2
MAA HYD 2
HYD MAA 2
GOI HYD 1
HYD GOI 1
COK HYD 1
HYD COK 1
BLR HYD 3
HYD BLR 3
DEL ATQ 2
ATQ DEL 2
BOM ATQ 1
ATQ BOM 1
MAA TRV 2
TRV MAA 2
COK TRV 2
TRV COK 2
BLR TRV 1
TRV BLR 1
HYD TRV 1
TRV HYD 1
BOM TRV 1

Auto-generated Files used:

The files which were edited/generated by the system are distributed into 2 folders which contain files similar to each other. The folders and the files are:

- Passengers: This is a folder which contains all the booking details of the passengers with reference to their passenger code.
  The file names are stored as: Pas<Passenger Code>.txt
  A sample file which was generated in the user manual is:
  File name: Pas100006.txt
  Contents:

  Date: 10-2-2021
  Source City: New Delhi
  Arrival City: Kolkata
  Is direct: True
  Flight Number: DF108

  Departure Time: 13:00
  Arrival Time: 15:00
  Total Cost: Rs6500
  Number of passengers: 1

  Details of Passengers:

Name	Age	Seat No.	Mobile Number	Gender
Shrivardhan Goenka	16	30C	9830376021	Male

- Data: This is a folder which contains the seats booked per flight and the seat numbers which are booked. There is a separate file for each day.
  The files are saved in this format: seats<dd>-<mm>-<yyyy>
  These files are very big so just a snippet which outlines the data for a single flight is:
  104
  180
  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 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 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 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  The first line is the flight number, the second line is the number of available seats on the flight, and the next 6 lines show the seats are which are empty(0 denotes empty and 1 denotes booked)