COEN – 6312: Model Driven Software Engineering

DELIVERABLE 1

FLIGHT RESERVATION SYSTEM

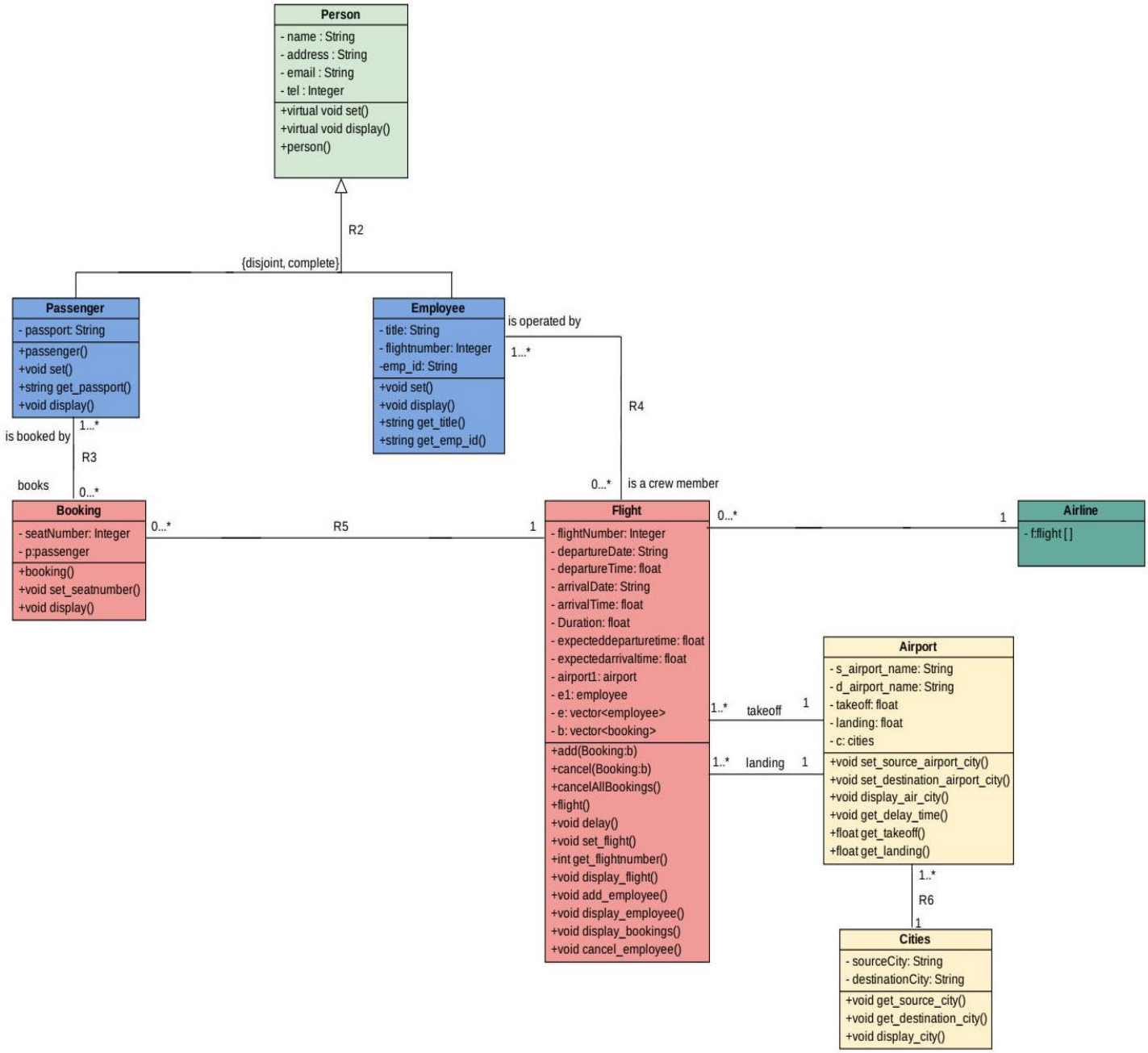
Submitted To:

Prof. Wahab Hamou-Lhadj

Submitted By:

Neerav Bhatia (40090577)
Bibin Eugine (40075231)

CLASS DIAGRAM



INSTRUCTIONS WITH EXAMPLE

The system can be access by two people: -

1. Airline employee id - air23 (id)
2. Agent id - ag21 (id)

PHASE – I (From Airline employee point of view)

Enter the necessary input function in the below mentioned order: -

A. Flight data
B. Employee
C. Passenger bookings (View only)
D. Delay (Applicable from Airline employee point of view)
E. CancelAllBooking
F. exit

A. Enter Flight data

i. Enter Airline employee ID/Agent ID: - air23 (Note: - At beginning, enter the airline employee id)

```
clang version 7.0.0-3~ubuntu0.18.04.1 (tags/RELEASE_700/final)
> clang++-7 -pthread -o main main.cpp
> ./main
Enter Airline employee ID/ Agent ID
air23
1. Flight information Add/View
2. Employee Add/Remove/view
3. Passenger information
4. Cancel All bookings
5. exit
```

Fig 1. Entering Airline ID in beginning

ii. Enter into the flight information add/view: (Add all the flights which will be taking/landing from source and destination cities. Fill out all the information, date: - DD/MM/YYYY time: - 23.05(means 11:05 pm)

```
3. Passenger information
4. Cancel All bookings
5. exit
1
Press 1 for viewing current list of flight
press 2 for adding new flights
press 3 for airline delay
2
Enter no of flights
2
Enter the details of flight: 1
Enter the flight number
123
Enter the departure date DD/MM/YYYY
01/02/2020
Enter the departure time
03.00
Enter the arrival date DD/MM/YYYY
01/02/2020
Enter the arrival time
06.00
Enter the duration
03.00
Enter the source airport name
YUL
Enter the source city
Montreal
Enter the destination airport name
IGI
Enter the destination city
New Delhi
Enter the details of flight: 2
Enter the flight number
987
Enter the departure date DD/MM/YYYY
30/03/2020
Enter the departure time
13.00
Enter the arrival date DD/MM/YYYY
24/04/2020
Enter the arrival time
16.00
Enter the duration
03.00
Enter the source airport name
Vacouver Int.
Enter the source city
Vancouver
Enter the destination airport name
Rajiv Gandhi Int. Airport
Enter the destination city
Hyderabad
Enter Airline employee ID/ Agent ID
```

Fig 2. Adding new flights

B. Enter Employee Details (Add all the employees who will be assigned to their respective flight numbers. One can be Crew or as a Manager) Check Figure 3. (**Note: title: Manager/crew, case sensitive)**

C. View the passenger information who will be travelling on different flights. Check Fig 4**. (Note: - First add the passengers' details by using Agent Id: ag21 since booking can be made by agent only. Then view the passenger information by using Agent ID: - air23)**

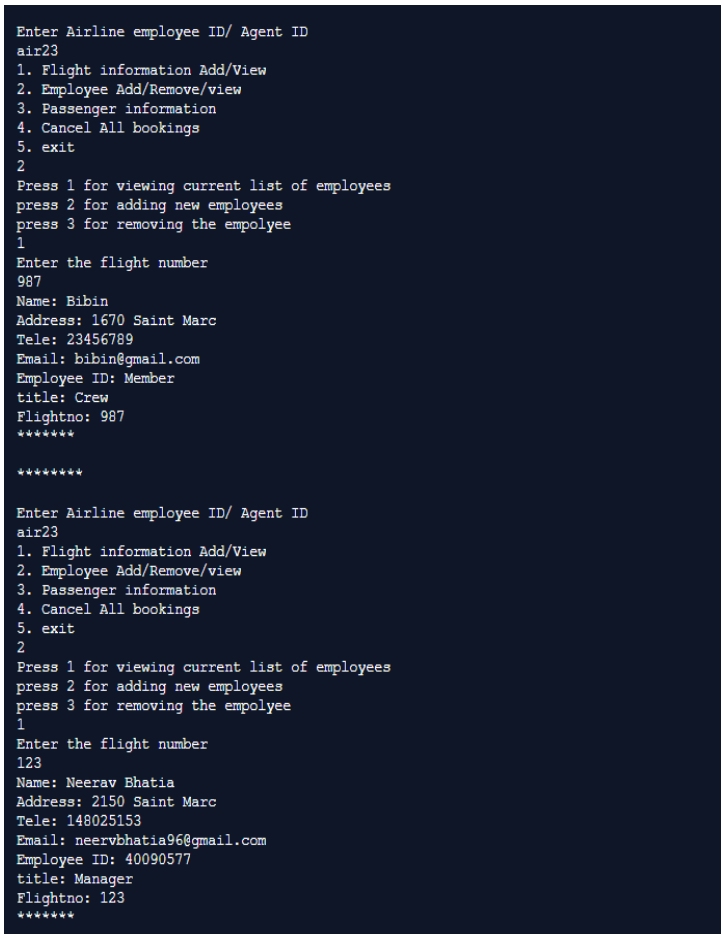


Fig 3. Adding Employee Details

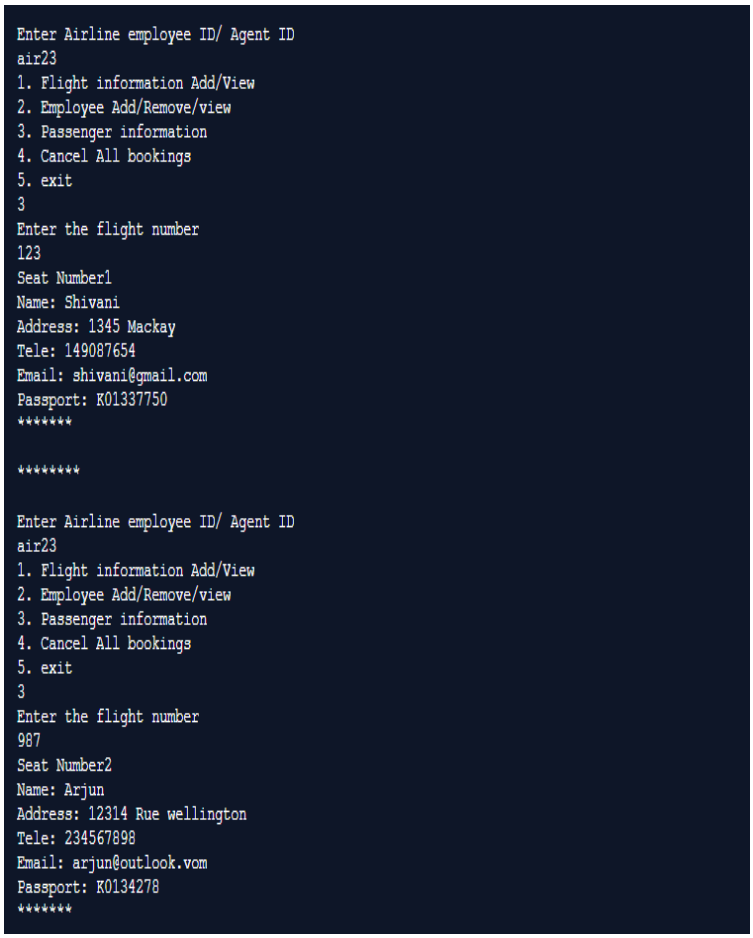


Fig 4. Viewing Passenger information

D. Add the delay as shown in Fig 5 (if flight is delayed, in DD/MM/YYYY format and time is in (23.05 format)

Flight No	**123**	**9876**
Departure	03.00	13.00
Arrival	06.00	16.00
Expected Departure	04.15	13.15
Expected Arrival	07.15	16.15
Delay	1 hr 15 mins	15 mins

E. CancelAllBooking as shown in Fig 6. (Note: Cancel all booking of passengers)

```
Press 1 for viewing current list of flight
press 2 for adding new flights
press 3 for airline delay
3
Enter the flight number
9876
Enter the takeoff delay
0.15
Enter the Landing delay
0.15
Enter Airline employee ID/ Agent ID
air23
1. Flight information Add/View
2. Employee Add/Remove/view
3. Passenger information
4. Cancel All bookings
5. exit
1
Press 1 for viewing current list of flight
press 2 for adding new flights
press 3 for airline delay
1
The details of flight: 1
Flight Number: 123
Departure Date: 01/02/2020
Departure Time: 3
Expected Departure Time: 4.15
Arrival Date: 01/02/2020
Arrival Time: 6
Expected Arrival Time: 7.15
Duration: 3
Source Airport: YUL
Destination Airport: IGI
Source City: Montreal
Destination City: New Delhi
*********

The details of flight: 2
Flight Number: 9876
Departure Date: 30/03/2020
Departure Time: 13
Expected Departure Time: 13.15
Arrival Date: 24/04/2020
Arrival Time: 16
Expected Arrival Time: 16.15
Duration: 3
Source Airport: Vancouver Int.
Destination Airport: Rajiv Gandhi Int.
Source City: Vancouver
Destination City: Hyderabad
*********
```

Fig 5. Delay of both the flights

```
Enter Airline employee ID/ Agent ID
air123
Wrong id, please enter again
Enter Airline employee ID/ Agent ID
air23
1. Flight information Add/View
2. Employee Add/Remove/view
3. Passenger information
4. Cancel All bookings
5. exit
4
Enter the flight number
123
********

Enter Airline employee ID/ Agent ID
air23
1. Flight information Add/View
2. Employee Add/Remove/view
3. Passenger information
4. Cancel All bookings
5. exit
3
Enter the flight number
123
********
```

Fig 6. Cancellation of all the flights

PHASE – II (From Agent point of view)

Enter the necessary input function in the below mentioned order: -

A. Enter Airline employee ID/Agent ID: - ag21
B. Book a flight (Enter Passenger Information)
C. Cancel Booking
D. exit

A. Enter Airline employee ID/Agent ID: - ag21
B. Book a flight for Passenger as shown in Fig 7.(Provide all information including passport id)

```
Enter Airline employee ID/ Agent ID
ag21
1. Book a flight
2. Cancel Booking
3. exit
1
Enter the flight number
123
Enter your name
Neerav
Enter your address
1290 Rue guy
Enter your telephone
71900971373
Enter your email
Neerv@gmail.com
Enter the passport number:
K017550
********
```

Fig 7. Booking a ticket for Passenger

C. Cancel Booking (By passport id)

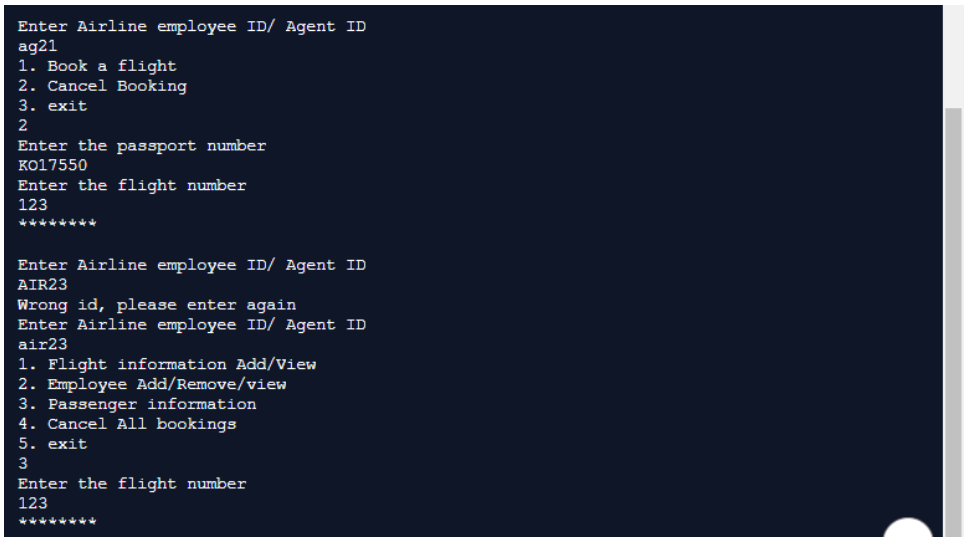


Fig 8. Cancelling a booking by passport number

D. exit

NOTE: Check Class diagram for getting idea about the data type of each variable.

SOURCE CODE

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

//class book declaration
class booking;

//Person class
class person{
private:
   string name;
   string address;
   int tel;
   string email;
public:
   person(string name = "nil", string address = "nil", int tel = 0, string email = "nil"){
      this->name = name;
      this->address = address;
      this->tel = tel;
      this->email = email;
   }

   virtual void set(){
      cout<<"Enter your name"<<endl;
      cin.ignore();
      getline(cin,name);
      cout<<"Enter your address"<<endl;
      getline(cin,address);
      cout<<"Enter your telephone"<<endl;
      cin.ignore();
      cin>>tel;
      cout<<"Enter your email"<<endl;
      cin>>email;
   }

   virtual void display(){
      cout<<"Name: "<<name<<endl;
      cout<<"Address: "<<address<<endl;
      cout<<"Tele: "<<tel<<endl;
      cout<<"Email: "<<email<<endl;
   }
   ~person()= default;
};
```

```
//passenger class
class passenger : public person{
private:
   string passport;
public:
   passenger(string passport = "nil"){
      this->passport = passport;
   };
   void set(){
      person::set();
      cout<<"Enter the passport number:"<<endl;
      //getline(cin,passport);
      cin>>this->passport;
   }
   string get_passport(){
      return passport;
   }
   void display(){
      person::display();
      cout<<"Passport: "<<passport<<endl;
   }
};

//employee class
class employee : public person{
private:
   string emp_id;
   string title;
   int flightno;
public:
   void set(){
      person::set();
      cout<<"Enter the title"<<endl;
      cin>>this->title;
      cout<<"Enter the Employee ID"<<endl;
      cin>>this->emp_id;
      cout<<"enter the flight number"<<endl;
      cin>>flightno;
   }
   void display(){
      person::display();
      cout<<"Employee ID: "<<emp_id<<endl;
      cout<<"title: "<<title<<endl;
      cout<<"Flightno: "<<flightno<<endl;
   }
   string get_title(){
      return  title;
```

```
    }
    int get_flightno(){
        return flightno;
    }
    string get_emp_id(){
        return emp_id;
    }
};

//Booking class
class booking{
     int seatnumber=0;
public:
    booking(int seatnumber = 0){
        this->seatnumber = seatnumber;
    }
    void set_seatnumber(){
        seatnumber++;
    }
    void display(){
        cout<<"Seat Number"<<seatnumber<<endl;
    }
    passenger p;
    ~booking()= default;
};

//Class Airports
class airport{
private:
    string s_airport_name, d_airport_name;
    float takeoff, landing;
public:
    class cities{
    private:
        string source_city, destination_city;
    public:
        void get_source_city(){
            cout<<"Enter the source city"<<endl;
            getline(cin, source_city);
        }
        void get_destination_city(){
            cout<<"Enter the destination city"<<endl;
            getline(cin, destination_city);
        }
        void display_city(){
            cout<<"Source City: "<<source_city<<endl;
            cout<<"Destination City: "<<destination_city<<endl;
        }
```

```
    };
    cities c;
    void set_source_airport_city(){
        cout<<"Enter the source airport name"<<endl;
        cin.ignore();
        getline(cin, s_airport_name);
         c.get_source_city();
    }
    void set_destination_airport_city(){
        cout<<"Enter the destination airport name"<<endl;
        getline(cin, d_airport_name);
        c.get_destination_city();
    }
    void display_air_city(){
        cout<<"Source Airport: "<<s_airport_name<<endl;
        cout<<"Destination Airport: "<<d_airport_name<<endl;
        c.display_city();
    }
    void get_delay_time(){
        cout<<"Enter the takeoff delay"<<endl;
        cin>>takeoff;
        cout<<"Enter the Landing delay"<<endl;
        cin>>landing;
    }
    float get_takeoff(){
        return takeoff;
    }
    float get_landing(){
        return landing;
    }
};

// Flight class
class flight{
private:
    int FlightNumber;
    string DepartureDate;
    float DepartureTime ;
    string ArrivalDate ;
    float ArrivalTime ;
    float duration ;
    float expe_dep_time;
    float expe_arr_time;
    vector<booking> b;
    vector<employee> e;
    employee e1;
    airport air;
public:
```

```
    flight(int flightno = 0, string depdate = "nil", float deptime = 0.00,
           string arrivaldate = "nil",float arrivaltime = 0.00,
           float duration = 0.0){
        this->FlightNumber = flightno;
        this->DepartureDate = depdate;
        this->DepartureTime = deptime;
        this->ArrivalDate = arrivaldate;
        this->ArrivalTime = arrivaltime;
        this->duration = duration;
    }
    void delay();
    void set_flight(){
        cout<<"Enter the flight number"<<endl;
        cin>>FlightNumber;
        cout<<"Enter the departure date DD/MM/YYYY"<<endl;
        cin.ignore();
        getline(cin,DepartureDate);
        cout<<"Enter the departure time"<<endl;
        cin>>DepartureTime;
        cout<<"Enter the arrival date DD/MM/YYYY"<<endl;
        cin.ignore();
        getline(cin,ArrivalDate);
        cout<<"Enter the arrival time"<<endl;
        cin>>ArrivalTime;
        cout<<"Enter the duration"<<endl;
        cin>>duration;
        air.set_source_airport_city();
        air.set_destination_airport_city();
        expe_arr_time = ArrivalTime;
        expe_dep_time = DepartureTime;
    }
    int get_flightno();
    void display_flight();
    void add_employee();
    void cancel_employee(string);
    void display_employee();
    void add(booking);
    void cancel(string);
    void CancelAllBookings();
    void display_bookings();
};

void flight ::delay() {
    air.get_delay_time();
    float temp;
    temp = (ArrivalTime + air.get_landing())- (int)((ArrivalTime + air.get_landing()));
    expe_arr_time = (int(ArrivalTime + air.get_landing()))%24+temp;
    temp = (DepartureTime + air.get_takeoff()) - (int)(DepartureTime + air.get_takeoff());
```

```
    expe_dep_time = (int(DepartureTime + air.get_takeoff()))%24+temp;
}

int flight ::get_flightno() {
    return FlightNumber;
}

void flight ::display_flight() {
    cout<<"Flight Number: "<<FlightNumber<<endl;
    cout<<"Departure Date: "<<DepartureDate<<endl;
    cout<<"Departure Time: "<<DepartureTime<<endl;
    cout<<"Expected Departure Time: "<<expe_dep_time<<endl;
    cout<<"Arrival Date: "<<ArrivalDate<<endl;
    cout<<"Arrival Time: "<<ArrivalTime<<endl;
    cout<<"Expected Arrival Time: "<<expe_arr_time<<endl;
    cout<<"Duration: "<<duration<<endl;
    air.display_air_city();
}

void flight::display_bookings(){
    vector<booking>::iterator it;
    booking temp;
    for(auto i = b.begin();i != b.end(); i++){
        temp = *i;
        temp.display();
        temp.p.display();
        cout<<"*******"<<endl<<endl;
    }
}

void flight::cancel(string p){
    vector<booking>::iterator it;
    booking temp;
    if(b.size()==1){
        b.clear();
    }else{
        for(auto it= b.begin(); it != b.end(); it++){
            temp = *it;
            if(temp.p.get_passport()==p){
                b.erase(it);
            }
        }
    }
}

void flight::CancelAllBookings() {
    b.clear();
}
```

```
void flight::add(booking b1) {
   booking temp;
   int count = 0;
   for(auto i = b.begin();i != b.end(); i++){
      temp = *i;
      if(temp.p.get_passport()==b1.p.get_passport())
         count =1;
   }
   if(count  == 0)
      b.push_back(b1);
   else
      cout<<"You cannot book more than one ticket for the same flight"<<endl;
}

void flight::add_employee() {
   e1.set();
   employee temp;
   int count =0;
   if(e1.get_title()=="Manager") {
      for (auto i = e.begin(); i != e.end(); i++) {
         temp = *i;
         if (temp.get_title() == "Manager") {
            count = 1;
         }
      }
   }
   if(count == 0 && (e1.get_flightno()==FlightNumber)){
      e.push_back(e1);
   }else{
      cout<<"Cannot assign more than one manager to a flight/ Wrong flight no "<<endl;
   }
}

void flight::cancel_employee(string id){
   vector<booking>::iterator it;
   employee temp;
   if(e.size()==1){
      e.clear();
   } else{
      for(auto it= e.begin(); it != e.end(); it++){
         temp = *it;
         if(temp.get_emp_id()==id){
            e.erase(it);
         }
      }
   }
}
```

```
void flight :: display_employee(){
   employee temp;
   for(auto i = e.begin();i != e.end(); i++){
      temp = *i;
      temp.display();
      cout<<"*******"<<endl<<endl;
   }
}

class airline{
public:
   flight f[5];
};

int main() {
   int k=1,n,n_emp,flight_no,no_of_flights=0;
   int temp = 0;
   string passport, airline_id = "air23", agent_id="ag21", id, empid;
   airline a;
   booking b;
   while(k){
      n_emp =0;
      temp =0;
      cout<<"Enter Airline employee ID/ Agent ID"<<endl;
      cin>>id;
      if(id == airline_id){
         cout<<"1. Flight information Add/View"<<endl;
         cout<<"2. Employee Add/Remove/view"<<endl;
         cout<<"3. Passenger information"<<endl;
         cout<<"4. Cancel All bookings"<<endl;
         cout<<"5. exit"<<endl;
         cin>>n;
         switch(n){
            case 1:cout<<"Press 1 for viewing current list of flight"<<endl;
                  cout<<"press 2 for adding new flights"<<endl;
                  cout<<"press 3 for airline delay"<<endl;
                  cin>>n_emp;
                  if(n_emp == 2) {
                    cout<<"Enter no of flights"<<endl;
                    cin>>no_of_flights;
                    for(int i=1; i<=no_of_flights; i++){
                       cout<<"Enter the details of flight: "<<i<<endl;
                       a.f[i].set_flight();
                    }
                  }else if(n_emp == 1) {
                     for(int i=1; i<=no_of_flights; i++){
                        cout<<"The details of flight: "<<i<<endl;
```

```
            a.f[i].display_flight();
            cout<<"*********"<<endl<<endl;
          }
        }else if(n_emp == 3){
          cout<<"Enter the flight number "<<endl;
          cin>>flight_no;
          for(int i=1; i<=no_of_flights; i++){
            if(a.f[i].get_flightno()==flight_no){
              a.f[i].delay();
              temp = 1;
            }
          }
          if(temp == 0)
            cout<<"invalid"<<endl;
        }else
          cout<<"Invalid"<<endl;

        break;
case 2:cout<<"Press 1 for viewing current list of employees"<<endl;
        cout<<"press 2 for adding new employees"<<endl;
        cout<<"press 3 for removing the empolyee"<<endl;
        cin>>n_emp;
        if(n_emp==1){
          cout<<"Enter the flight number "<<endl;
          cin>>flight_no;
          for(int i=1; i<=no_of_flights; i++){
            if(a.f[i].get_flightno()==flight_no){
              a.f[i].display_employee();
              cout<<"********"<<endl<<endl;
              temp = 1;
            }
          }
          if(temp == 0)
            cout<<"invalid"<<endl;
        }else if(n_emp==2){
          cout<<"Enter the flight number to which you need to assign"<<endl;
          cin>>flight_no;
          for(int i=1; i<=no_of_flights; i++){
            if(a.f[i].get_flightno()==flight_no){
              a.f[i].add_employee();
              cout<<"********"<<endl<<endl;
              temp = 1;
            }
          }
          if(temp == 0)
            cout<<"invalid"<<endl;
        }else if(n_emp ==3){
          cout<<"Enter the employee id"<<endl;
```

```
            cin>>empid;
            cout<<"Enter the flight number"<<endl;
            cin>>flight_no;
            for(int i=1; i<=no_of_flights; i++){
              if(a.f[i].get_flightno()==flight_no){
                a.f[i].cancel_employee(empid);
                temp = 1;
              }
            }
            if(temp == 0)
              cout<<"invalid"<<endl;
          }else
            cout<<"invalid"<<endl;
          break;
      case 3:cout<<"Enter the flight number"<<endl;
          cin>>flight_no;
        for(int i=1; i<=no_of_flights; i++){
          if(a.f[i].get_flightno()==flight_no){
            a.f[i].display_bookings();
            cout<<"********"<<endl<<endl;
            temp = 1;
          }
        }
        if(temp == 0)
          cout<<"invalid"<<endl;
        break;
      case 4:cout<<"Enter the flight number"<<endl;
        cin>>flight_no;
        for(int i=1; i<=no_of_flights; i++){
          if(a.f[i].get_flightno()==flight_no){
            a.f[i].CancelAllBookings();
            cout<<"********"<<endl<<endl;
            temp = 1;
          }
        }
        if(temp == 0)
          cout<<"invalid"<<endl;
        break;
      case 5:break;
      default:break;
    }
  }else if(id == agent_id){
    cout<<"1. Book a flight"<<endl;
    cout<<"2. Cancel Booking"<<endl;
    //cout<<"3. CancelALLbooking"<<endl;
    cout<<"3. exit"<<endl;
    cin>>n;
    switch(n){
```

```
            case 1:cout<<"Enter the flight number"<<endl;
                  cin>>flight_no;
                  b.set_seatnumber();
                  b.p.set();
                for(int i=1; i<=no_of_flights; i++){
                  if(a.f[i].get_flightno()==flight_no){
                     a.f[i].add(b);
                     cout<<"********"<<endl<<endl;
                     temp = 1;
                  }
                }
                if(temp == 0)
                   cout<<"invalid"<<endl;
                  break;
            case 2:cout<<"Enter the passport number"<<endl;
                  cin>>passport;
                  cout<<"Enter the flight number"<<endl;
                  cin>>flight_no;
                for(int i=1; i<=no_of_flights; i++){
                  if(a.f[i].get_flightno()==flight_no){
                     a.f[i].cancel(passport);
                     cout<<"********"<<endl<<endl;
                     temp = 1;
                  }
                }
                if(temp == 0)
                   cout<<"invalid"<<endl;
                  break;
            case 3:break;
            default:break;
          }
        }else
          cout<<"Wrong id, please enter again"<<endl;
    }
    return  0;
}
```