Sri Lanka Institute of Information Technology

Topic : Airline Ticket Reservation System.

Group Number : MLB_PG.06.02_12

Campus : Malabe

Submission Date : 19/10/2020

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT20077624	P.A. Daham Thameera	0711106045
IT20074340	Nirmal M.D.S	0767278015
IT20067038	Nirosh Dilranga	0762174135
IT20069018	Akila Wijewardane	0770892677
IT20010430	Ishadi Shashikala	0767447421

Exercise 1:

1) User Requirements

1. An Unregistered user in an online Airline Ticket Reservation system needs to first register providing details such as name, passport number, e-mail address.
2. Then the Registered user can log in to the system using his/her credentials.
3. If there are some mistakes in the registration details the registered user can edit user details.
4. A Registered User can choose the flight number, flight time, and location from the flight schedule.
5. A Registered User can enter reservation details such as class type (Business / Economy), food type, seat number.
6. The Registered User can view the status of the reservation details.
7. The Registered User can edit the reservations from the cart.
8. The Registered User selects the payment method (credit card, debit card) and enters payment details for the reservation.
9. The Registered User confirms the reservation and the payment is validated the reservation and the serial codes of tickets for each seat number are updated.
10. The Registered User views the serial codes of tickets according to the reservation that they made.
11. The Support Agent log in to the system.
12. The Support Agent generates a list of reservations and checks the reservation details.
13. The Support Agent uploads the air ticket after processing to the system according to the list of reservations that the Support Agent generates and the Registered User can view the ticket after uploading it.
14. The online Airline Ticket administrator can add new flight schedules, seat numbers, class types according to the available tickets that the system has.
15. The online Airline Ticket administrator gets a list of ticket serial numbers of the seat numbers that have already been reserved and update the system.

2) Noun & verb Analysis

(Noun are in red and Verb are in blue),

1. An Unregistered user in an online Airline Ticket Reservation system needs to first register providing details such as name, passport number, e-mail address.
2. Then the Registered user can log in to the system using his/her credentials.
3. If there are some mistakes in the registration details the registered user can edit user details.
4. A Registered User can choose the flight number, flight time, and location from the flight schedule.
5. A Registered User can enter reservation details such as class type (Business / Economy), food type, seat number.
6. The Registered User can view the status of the reservation details.
7. The Registered User can edit the reservations from the cart.
8. The Registered User selects the payment method (card/paypal) and enters payment details for the reservation.
9. The Registered User confirms the reservation and the payment is validated the reservation and the serial codes of tickets for each seat number are updated.
10. The Registered User views the serial codes of tickets according to the reservation that they made.
11. The Support Agent log in to the system.
12. The Support Agent generates a list of reservations and checks the reservation details.
13. The Support Agent uploads the air ticket to the system according to the list of reservations that the Support Agent generates and the Registered User can view the ticket after uploading it.
14. The online Airline Ticket administrator can add new flight schedules, seat numbers, class types according to the available tickets that the system has.
15. The online Airline Ticket administrator gets a list of ticket serial numbers of the seat numbers that have already been reserved and update the system.

3) Identified classes using Noun Analysis,

Identified classes;

- User
- Flight
- Reservation
- Payment
- Card - (Inherited from Payment)
- Paypal - (Inherited from Payment)
- Cart
- Ticket
- Agent
- Report

Reasons for rejecting other nouns

1. **Redundant:**

    - Registered user/Unregistered user**->** refers to the same person "User"
    - Flight Schedule **->** refers to the "Flight"
    - Support Agent **->** refers to the "Agent"

2. **Outside the Scope:**

    - System
    - online Airline Ticket administrator

3. **Attributes:**

    - Name, Passport number, e-mail address.
    - flight number, flight time, and location.
    - class type (Business / Economy), food type, seat number.
    - seat number
    - registration details

Exercise 2:

CRC Cards for the online Airline Ticket Reservation system

User	
Responsibility	**Collaborators**
Store Details of User	
Edit User Details	
Log in	

Flight	
Responsibility	**Collaborators**
Store Flight Details	
Add flight schedules	Ticket

Reservation	
Responsibility	**Collaborators**
Store Reservation Details	
Status of Reservation	
Add Reservation Details	Ticket
Update reserved seats number	Ticket
Confirm Reservation	Payment

Payment	
Responsibility	**Collaborators**
Store Payment Details	
Validate	

Card	
Responsibility	**Collaborators**
Store Card Payment Details	Payment

Paypal	
Responsibility	**Collaborators**
Store Paypal Payment Details	Payment

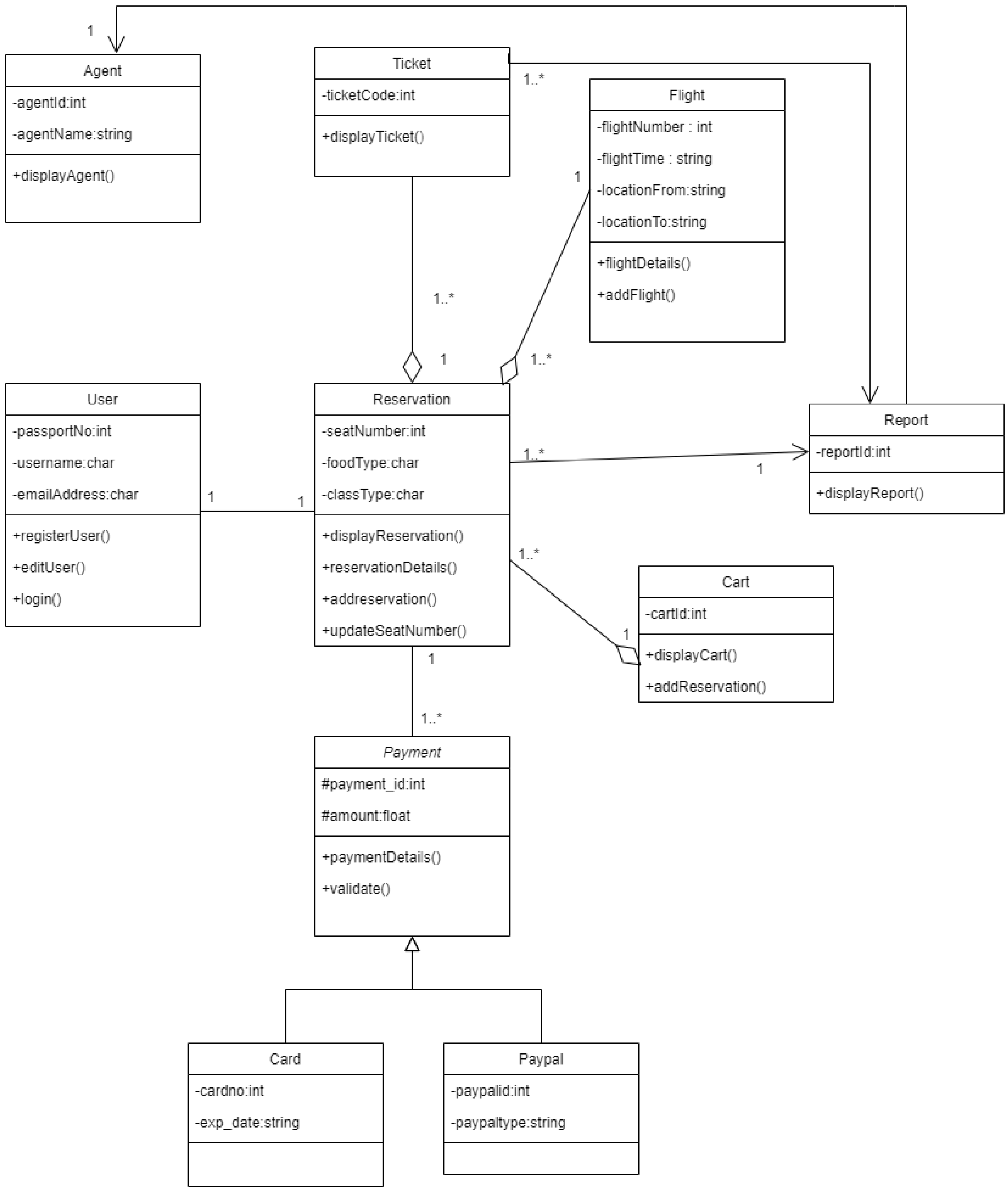
Cart	
Responsibility	**Collaborators**
Edit Reservations	Reservation

Ticket	
Responsibility	**Collaborators**
Update Serial Codes of Tickets	
Display Serial Codes of Tickets	
Display ticket	Reservation

Report	
Responsibility	**Collaborators**
List of Reservations	Reservation
list of ticket serial numbers	Ticket

Exercise 3:

Class diagram for the Airline Reservation System



Exercise 4:

C++ Coding

```
#include <iostream>
#include<cstring>
#include<string>
using namespace std;
#define SIZE 2

//User class
class User{
 private:
   int passportno;
   char username[20];
   char emailAddress[];

 public:
   User();
   User(int ppassportno, char pusername[],char pemailAddress[]);
   void registeruser();
   void edituser();
   void login();
};

//Flight class
class Flight{
 private:
   int flightNumber;
   string flightTime;
   string locationFrom;
   string locationTo;

 public:
   Flight();
   Flight(int pflightNumber, string pflightTime, string plocationFrom, string
plocationTo);
   void flightDetails();
   void addflight();
};
```

```
//Reservation class
class Reservation{
 private:
  Flight *fli[SIZE];
  int seatNumber;
  string foodType;
  string classType;

 public:
  Reservation();
  Reservation(int pseatNumber, string pfoodType,string pclasstype);
  void addFlight(Flight *fli1, Flight *fli2);
  void reservationDetails();
  void displayReservation();
  void addreservation();
  void updateSeatNumber();
};

//Cart Class
class Cart{
 private:
  Reservation *res[SIZE];
  int cartid;

 public:
  Cart();
  Cart(int pcartid);
  void addReservation(Reservation *res1, Reservation *res2);
  void displayCart();
};

//Agent class
class Agent
{
 private:
  int agentid;
  string agentName;

 public:
  Agent();
  Agent(int pagentId, string pagentName);
  void displayAgent();
};
```

```
//Report class
class Report{
 private:
  int reportid;
  Agent *age;

 public:
  Report();
  Report(int preportid, Agent *a);
  void displayReport();
};

//Ticket class
class Ticket{
 private:
  int ticketCode;
  Report *rep;

 public:
  Ticket();
  Ticket(int pticketCode, Report *r);
  void displayTicket();
};

//Payment class
class Payment{
 protected:
   int payment_id;
   float amount;

 public:
   Payment();
   Payment(int ppayment_id,float pamount);
   void paymentDetails();
   void validate();


};
```

```
//Card class
class Card : public Payment{
 private:
   int cardno;
   string expDate;

 public:
   Card();
   Card(int pcardno, string pexpDate);

};

//Paypal class
class Paypal : public Payment{
 private:
   int paypalid;
   string paypalCode;

 public:
   Paypal();
   Paypal(int ppaypalid,string ppaypalCode);
};
```

```
//method implementation

//User class method implementation
User::User(){}

User:: User(int ppassportno, char pusername[],char pemailAddress[]){
  passportno = ppassportno;
  strcpy(username,pusername);
  strcpy(emailAddress,pemailAddress);
}
void User :: registeruser(){}
void User :: edituser(){}
void User :: login(){}

//Flight implementation
Flight::Flight(){}

Flight::Flight(int pflightNumber, string pflightTime, string plocationFrom,
string plocationTo){
  flightNumber = pflightNumber;
  flightTime = pflightTime;
  locationFrom = plocationFrom;
  locationTo = plocationTo;
}
void Flight::addflight(){}
void Flight::flightDetails(){}

//Reservation implementation
Reservation::Reservation(){}
Reservation::Reservation(int pseatNumber,string pfoodType,string pclasstype){
  seatNumber = pseatNumber;
  foodType = pfoodType;
  classType = pclasstype;
}
void Reservation::reservationDetails(){}

void Reservation::addFlight(Flight *fli1, Flight *fli2)
{
  fli[0] = fli1;
  fli[1] = fli2;
}
```

```
void Reservation::displayReservation()
{
  cout << "Seet Number : " << seatNumber << endl;
  cout << "Food Type : " << foodType << endl;
  cout << "Class Type : " << classType << endl << endl;
}
void Reservation:: addreservation(){}
void Reservation:: updateSeatNumber(){}

//Cart implementation
  Cart::Cart(){}
  Cart::Cart(int pcartid)
  {
    cartid = pcartid;
  }

  void Cart::addReservation(Reservation *res1, Reservation *res2)
  {
    res[0] = res1;
    res[1] = res2;
  }

//Agent implementation
Agent::Agent(){}
Agent::Agent(int pagentId, string pagentName)
  {
    agentid = pagentId;
    agentName = pagentName;
  }

  void Agent::displayAgent()
  {
    cout << "Agent ID : " << agentid << endl;
    cout << "Agent Name : " << agentName << endl;
  }
```

```
//Report implementation
 Report::Report(){}
 Report::Report(int preportid, Agent *a)
  {
    reportid = preportid;
    age = a;
  }

  void Report::displayReport()
  {
   cout << "Report ID : " << reportid << endl;
   age->displayAgent();
  }

//Ticket implementation
Ticket::Ticket(int pticketCode, Report *r)
  {
    ticketCode = pticketCode;
    rep = r;
  }

  void Ticket::displayTicket()
  {
    cout << "Ticket Code : " << endl;
    rep->displayReport();
  }

//Payment implementation
Payment::Payment(){}
Payment:: Payment(int ppayment_id,float pamount){
  payment_id = ppayment_id;
  amount = pamount;
}
void Payment::paymentDetails(){}
void Payment::validate(){}

//Card implementation
Card::Card(){}
Card::Card(int pcardno, string pexpDate){
  cardno = pcardno;
  expDate = pexpDate;
}
```

```
//Paypal implementation
Paypal::Paypal(){}
Paypal::Paypal(int ppaypalid,string ppaypalCode){
  paypalid = ppaypalid;
  paypalCode = ppaypalCode;
}

//Main Program
int main() {

  User u1(123456789,"Sunimal","sunimal@smail.com");
  Flight f1(441,"1100h","SriLanka","Japan");
  Reservation r1(47,"Asian","Business");
  Payment p1(1004,140000.00);
  Card c1(987654321,"23/08");
  Paypal pal1(100040008,"Personal");
  Cart cart1;
  Agent a1;
  Report rep1;

  return 0;

  //std::cout << "Hello World!\n";
}
```