Topic : Airline Ticket Reservation

Group no: MLB_01.02_12

Campus: Malabe

Submission Date: 10/15/2021

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**
IT21061134	Samarahewa N. Y.	0769713829
IT21104244	Bandara I. K. T.	0777894491
IT21053832	Wickramarachchi D. C.	0769308802
IT21096952	Gunaratna K. M. S. M.	0766262357
IT21103940	Ekanayake E. M. D. S. B.	0717701206

Exercise 1

- ### User Requirements.

1. Unregistered Customer of airline ticket reservation system needs to register providing details such as name, e-mail address, password.
2. Unregistered Customer can search flights by entering departure airport, destination airport, and the travel dates.
3. Registered Customer can book a ticket by entering relevant details.
4. Registered Customer can choose a payment method for booking. e.g., credit card, debit card, google pay.
5. Registered Customer can see the details of the tickets booked.
6. Registered Customer can change the date of the ticket.
7. Registered Customer can cancel ticket. If cancellation deadline is exceeded customer must pay additional cancellation fee.
8. Registered Customer can request for refunds on cancelled tickets.
9. Registered user can view the purchase history.
10. Registered user can update user profile.
11. System administrator can add new flights, modify the details of existing flights, and remove flights from the system.
12. System Administrator can see refund requests.
13. System Administrator can see details of the customer’s bookings.
14. A customer can also register as a Travel agent.
15. If Travel Agent booked more than 25 tickets within a month he/she can received 25% discount.

- **Noun Analysis.**

01. Unregistered Customer of airline ticket reservation system needs to register providing details such as name, e-mail address, password.
02. Unregistered Customer can search flights by entering departure airport, destination airport, and the travel dates.
03. Registered Customer and Travel agent can book a ticket by entering relevant details.
04. Registered Customer and Travel agent can choose a payment method for booking. e.g. credit card, debit card, google pay.
05. Registered Customer and Travel agent can see the details of the tickets booked.
06. Registered Customer and Travel agent can change the date of the ticket.
07. Registered Customer and Travel agent can cancel ticket. If cancellation deadline is exceeded customer has to pay additional cancellation fee.
08. Registered Customer and Travel agent can request for refunds on cancelled tickets.
09. Registered user can view the purchase history.
10. Registered user can update user profile.
11. System administrator can add new flights, modify the details of existing flights and remove flights from the system.
12. System Administrator can see refund requests.
13. System Administrator can see details of the customer's bookings.
14. A customer can also register as a Travel agent.
15. If Travel Agent booked more than 25 tickets within a month he/she can received 25% discount.

Classes.

- Travel Agent.
- Registered Customer.
- Unregistered Customer.
- Ticket.
- Payment.
- Flight.

Attributes.

- Name.
- E-mail address.
- Password.

Redundant.

- Credit card.
- Debit card.
- Google pays.

Outside the scope.

- System.
- System Administrator

Exercise 2.

- **CRC Cards.**

Customer	
Responsibility	**Collaborators**
Book Flight Ticket	Flight, Ticket
Make Payment	Payment
Search Flight	Flight

Registered Customer	
Responsibility	**Collaborators**
Send Feed back	
Change dates	Ticket

Unregistered Customer	
Responsibility	**Collaborators**
Create an account	
Search flights	Flight
Search for facilities	

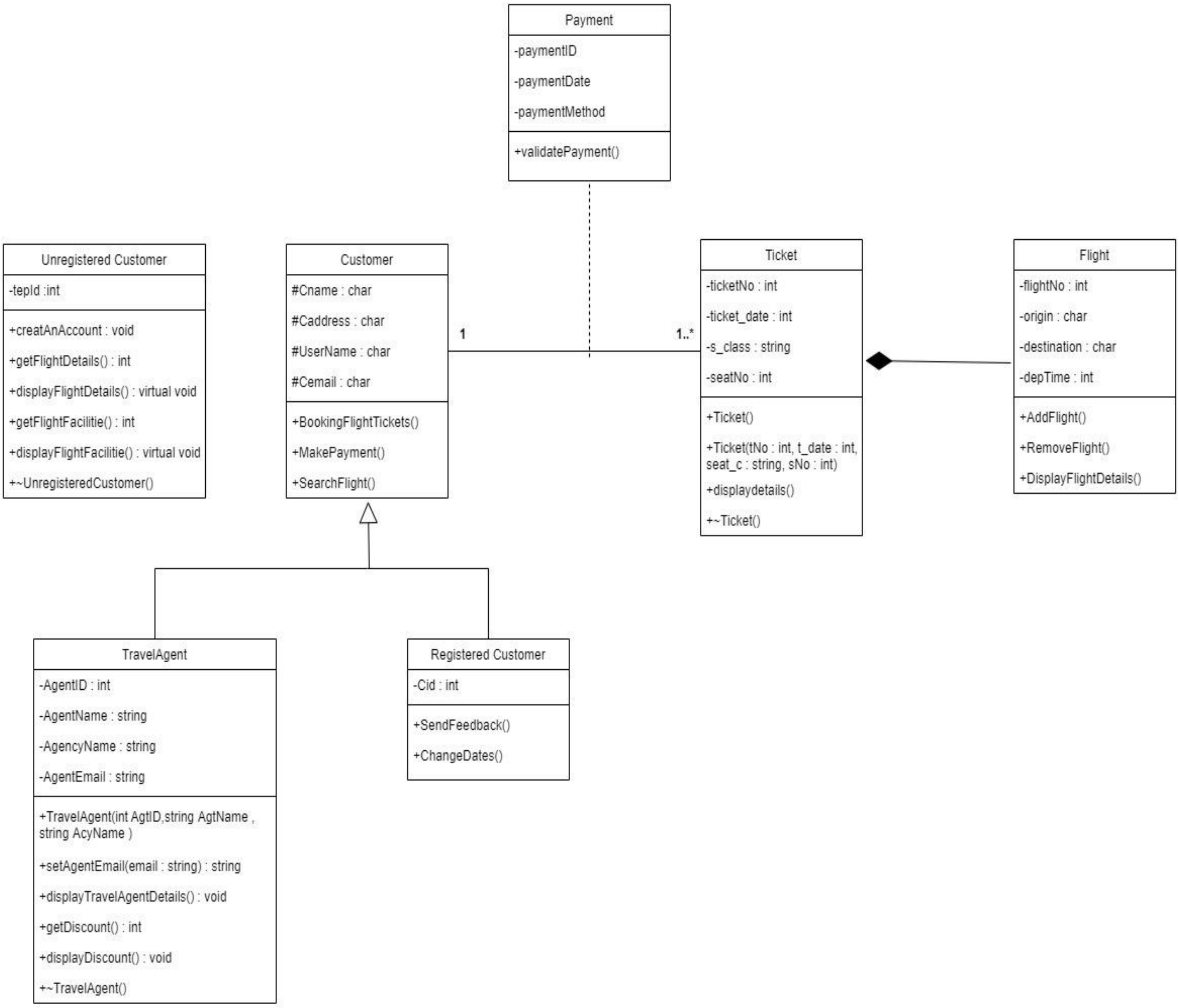
<table>
<tr><th colspan="2">Travel Agent</th></tr>
<tr><th>Responsibility</th><th>Collaborators</th></tr>
<tr><td>Keep Travel Agent Details</td><td></td></tr>
<tr><td>Managing Discount</td><td></td></tr>
<tr><td>Update Travel Agent Details</td><td></td></tr>
<tr><td>Display Travel Agent Details</td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Flight</th></tr>
<tr><th>Responsibility</th><th>Collaborators</th></tr>
<tr><td>Add Flights</td><td></td></tr>
<tr><td>Modify flight details</td><td></td></tr>
<tr><td>Remove Flights</td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Ticket</th></tr>
<tr><th>Responsibility</th><th>Collaborators</th></tr>
<tr><td>Making the reservation</td><td>Flight</td></tr>
<tr><td>Verifying Payment</td><td>Payment</td></tr>
<tr><td></td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Payment</th></tr>
<tr><th>Responsibility</th><th>Collaborators</th></tr>
<tr><td>Store payment details</td><td>Flight, Ticket, Customer</td></tr>
<tr><td></td><td></td></tr>
</table>

Exercise 3.



- **Class Diagram.**

Exercise 4.

- **Code.**

  Header Files

customer .h

/*customer*/

```
#include <iostream>
#include "Ticket.h"
using namespace std;
#define SIZE 10;
class Customer
{

protected:
    char Cname[30];
    char Caddress[50];
    char userName[30];
    char Cemail[30];
    Ticket * ticket[SIZE];



public:
    Customer();
    Customer(char name[], char address[], char uName[], char email[]);
    void bookingFlightTicket();
    void makePayment();
    void sreachFlight();
    ~Customer();
};
```

```
/*Flight*/
#include <iostream>
using namespace std;

class Flight {

private:

int flightNo;

char origin[50];

char destination[50];

int depTime;

public:

Flight();

Flight(int no, const char porigin[], const char pDestination[], int
pDepTime);

void DisplayFlightDetails();

~Flight();

};



/*Payment*/

#include <iostream>
#include <cstring>
using namespace std;
```

```
class Payment
{
private:
int paymentID;
int paymentDate;
string paymentMethod;

public:
Payment();
Payment(int id, int pDate, string pMethod);
void validatePayment();

~Payment();
};
```

```
/*RegisteredCustomer.h*/

#include <iostream>
#include "Customer.h"
using namespace std;
```

```cpp
class RegisteredCustomer : public Customer
{

private:
    int Cid;

public:
    RegisteredCustomer();
    RegisteredCustomer(int cid, char name[], char address[], char uName[],
char email[]);
    void SendFeedback();
    void ChangeDates();
    ~RegisteredCustomer();
};
```

```cpp
/*Ticket*/

#include <iostream>
#include <cstring>
using namespace std;


class Ticket
```

```
{
private:
int ticketNo;
int ticket_date;
string s_class;
int seatNo;
public:
Ticket();
Ticket(int tNo, int t_date, string seat_c,int sNo);
void Displaydetails();

~Ticket() ;
};
```

```
/*TravelAgent*/

#include <iostream>
#include <cstring>
using namespace std;

class TravelAgent : public Customer
{
```

```
private:
int AgentID;
string AgentName;
string AgencyName;
string AgentEmail;



public:
TravelAgent();
TravelAgent(int AgtID,string AgtName ,string AcyName );
void setAgentEmail(string email);
void displayTravelAgentDetails();
int getDiscount();
void displayDiscount();
~TravelAgent();




};
```

```
/*Unregistered Customer*/
#include <iostream>
#include <cstring>

using namespace std;


class UnregisteredCustomer
{
```

```
private:
int tepId;
public:
UnregisteredCustomer();
UnregisteredCustomer(int id);
void creatAnAccount();
int getFlightDetails();
virtual void displayFlightDetails();
int getFlightFacilitie();
virtual void displayFlightFacilitie();
~UnregisteredCustomer();
};
```

CPP Files

```
/* Customer.cpp*/


#include <iostream>
#include <string.h>
#include "Customer.h"

Customer::Customer()
```

```
{
}
Customer::Customer(char name[], char address[], char uName[], char email[])
{
    strcpy(Cname, name);
    strcpy(Caddress, address);
    strcpy(userName, uName);
    strcpy(Cemail, email);
}
void Customer::bookingFlightTicket()
{
}
void Customer::makePayment()
{
}
void Customer::sreachFlight()
{
}

Customer::~Customer()
{
}
```

/*Flight*/

```
#include "Flight.h"
#include <cstring>


Flight::Flight()
{
flightNo=0;

strcpy(origin,"");

strcpy(destination,"");

depTime=0;

}
```

```
Flight::Flight(int no, const char porigin[], const char pDestination[],
int pDepTime)
{
flightNo=no;

strcpy(origin,porigin);

strcpy(destination,pDestination);

depTime=pDepTime;
}

void Flight::DisplayFlightDetails()
{

}

Flight::~Flight()
{
}






/*Payment*/

#include "Payment.h"
Payment::Payment()
{
int paymentID;
int paymentDate;
string paymentMethod;
}


Payment::Payment(int id, int pDate, string pMethod) {
paymentID = id;
paymentDate=pDate;
paymentMethod=pMethod;
}

void Payment::validatePayment() {
}
Payment::~Payment()
{
}
```

```
/*Registered Customer.cpp*/


#include <iostream>
#include "RegisteredCustomer.h"

RegisteredCustomer ::RegisteredCustomer()
{
    Cid = 0;
}

RegisteredCustomer ::RegisteredCustomer(int cid, char name[], char
address[], char uName[], char email[]) : Customer(name, address, uName,
email)
{
    Cid = cid;
}

void RegisteredCustomer ::SendFeedback()
{
}

RegisteredCustomer ::~RegisteredCustomer()
```

```
{
}#include <iostream>
#include "RegisteredCustomer.h"

RegisteredCustomer ::RegisteredCustomer()
{
    Cid = 0;
}

RegisteredCustomer ::RegisteredCustomer(int cid, char name[], char
address[], char uName[], char email[]) : Customer(name, address, uName,
email)
{
    Cid = cid;
}

void RegisteredCustomer ::SendFeedback()
{
}

RegisteredCustomer ::~RegisteredCustomer()
{
}
```

```
/*Tikcet*/
```

```
#include "Ticket.h"

Ticket::Ticket()
{
int ticketNo=0;
int ticket_date=0;
string s_class="";
int seatNo=0;
}


Ticket::Ticket(int tNo, int t_date, string seat_c,int sNo) {
ticketNo = tNo;
ticket_date = t_date;
s_class = seat_c;
seatNo = sNo;
}

void Ticket::Displaydetails(){
}
Ticket::~Ticket()
{
}
```

```
/*TravelAgent*/
```

```
#include "TravelAgent.h"

TravelAgent::TravelAgent()
{
}
TravelAgent::TravelAgent(int AgtID,string AgtName ,string AcyName )
{
AgentID=AgtID;
AgentName=AgtName;
AgencyName=AcyName;


}
void TravelAgent::setAgentEmail(string email)
{
}
void TravelAgent::displayTravelAgentDetails()
{
}
int TravelAgent::getDiscount()
{
}
void TravelAgent::displayDiscount()
{
}
TravelAgent::~TravelAgent()
{
}
```

```
/*UnregisteredCustomer*/
```

```
#include "UnregisteredCustomer.h"

UnregisteredCustomer::UnregisteredCustomer(int id)
{
tepId=id;
}


void UnregisteredCustomer::creatAnAccount()
{

}
int UnregisteredCustomer::getFlightDetails()
{

}
void UnregisteredCustomer::displayFlightDetails()
{

}
int UnregisteredCustomer::getFlightFacilitie()
{

}
void UnregisteredCustomer::displayFlightFacilitie()
{

}
UnregisteredCustomer::~UnregisteredCustomer()
{

}
```

Main CPP

```
#include <iostream>
#include <cstring>
#include "Customer.h"
#include "TravelAgent.h"
#include "Flight.h"
#include "Payment.h"
#include "RegisteredCustomer.h"
#include "Ticket.h"
#include "UnregisteredCustomer.h"
using namespace std;
int main()
{
Flight*fli=new Flight();
Payment*pay =new Payment();
RegisteredCustomer*regcustomer =new RegisteredCustomer();
UnregisteredCustomer*unrcustomer=new UnregisteredCustomer();
TravelAgent*travag=new TravelAgent();
Ticket*tick= new Ticket();

delete fli;
delete pay;
delete regcustomer;
delete unrcustomer;
delete travag;
delete tick;



return 0;
}
```