Object Oriented Concept – IT1050

Topic : Airline Ticket Reservation.

Group No : MLB_WD_CSNE_02.01_02

Campus : Malabe

Submission Date : 14.06.2023

We declare that this is our own work, and this Assignment does not incorporate 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.

Name
Vijayananthan.D
Jayanath D.T
Sharvajen.S
Harikirushna.T
Pakeershan.C

Table of Contents

System requirements:

- All users can access the website.
- Users can register the website by providing user ID, name, and address, date of birth, phone number, gender and email.
- Users can view the website's content, including promotions, customer feedback, available flight packages, and frequently asked questions (FAQs).
- Visitors can register for the website to become registered users.
- During registration, users must provide their name, username, address, and password, date of birth, phone number, gender, email.
- Registered users can make and cancel reservations by providing their user name and flight name.
- Registered users can add inquiries by providing an inquiry ID, phone no, and email.
- Registered user can edit their profiles and add feedback about their flight by providing a user ID.
- Manage the reservation process, including flight and seat selection.
- Verify passenger information and confirm the proper ticket.
- Give the information about their reservation.
- Registered members can provide feedback.
- They can share their experiences and offer suggestions or comments to help improve the services.
- The administration analyzes feedback to identify areas for improvement.
- Administrators have privileged access to the website.
- They can log in using a username and password.
- Administrators have the authority to approve or decline reservations made by members.
- They can activate or deactivate user accounts as necessary.
- They handle user accounts, including registration, login, and authentication.
- Handle system maintenance and updates.
- Process payment securely.
- Collaborate with external payment gateways and financial institutions.
- Handle transaction verifications and confirmations.

- Update ticket status.
- Managers have specific access privileges to perform managerial tasks.
- They oversee the airline ticket reservation system.
- They can generate reports related to reservations, feedback, and inquiries.
- Generate reports related to system performance, booking statistics, feedback, inquiry, etc.
- Provide analysis based on collected data.
- Registered members can make inquiries by providing their phone number and email address.
- This allows members to seek assistance or gather additional information.
- Receive and address inquiries from users regarding reservations, flights, or other system-related information.
- Provide accurate and timely responses to user inquiries.

Classes

Based on the provided description, the following classes can be identified for the FlyEasy ticket reservation system:

- User
- Registered User
- Flight Reservation
- Feedback
- Administration
- Payment.
- Manager
- Report
- Inquiries

CRC CARD

USER	
Responsibilities	collaborations
Create an account	
Access the website	Registered user
Search for flights and dates	
Make payments for ticket purchases	manager
Provide feedback	feedback
Make inquiries	inquiries

Registered User	
Responsibilities	collaborators
Make flight reservation	Flight reservation
Cancel flight reservation	
Manage account settings and profile information	
Share travel experience	
Provide feedback	Feedback
Make inquiries	inquiry
Manage account experience	

Flight reservation	
Responsibilities	collaborators
Make a reservation	User
Cancel a reservation	Flight reservation
Get reservation status	
Manage availability and pricing of flight	

Feedback	
Responsibilities	Collaborators
Provide feedback	User
Analyze feedback	administration

Administration	
Responsibilities	collaborators
Manage reservation	Flight reservation
Handle user account	user
Handle system maintenance and updates	
Manage staffs details	
Implement improvements using feedback	feedback

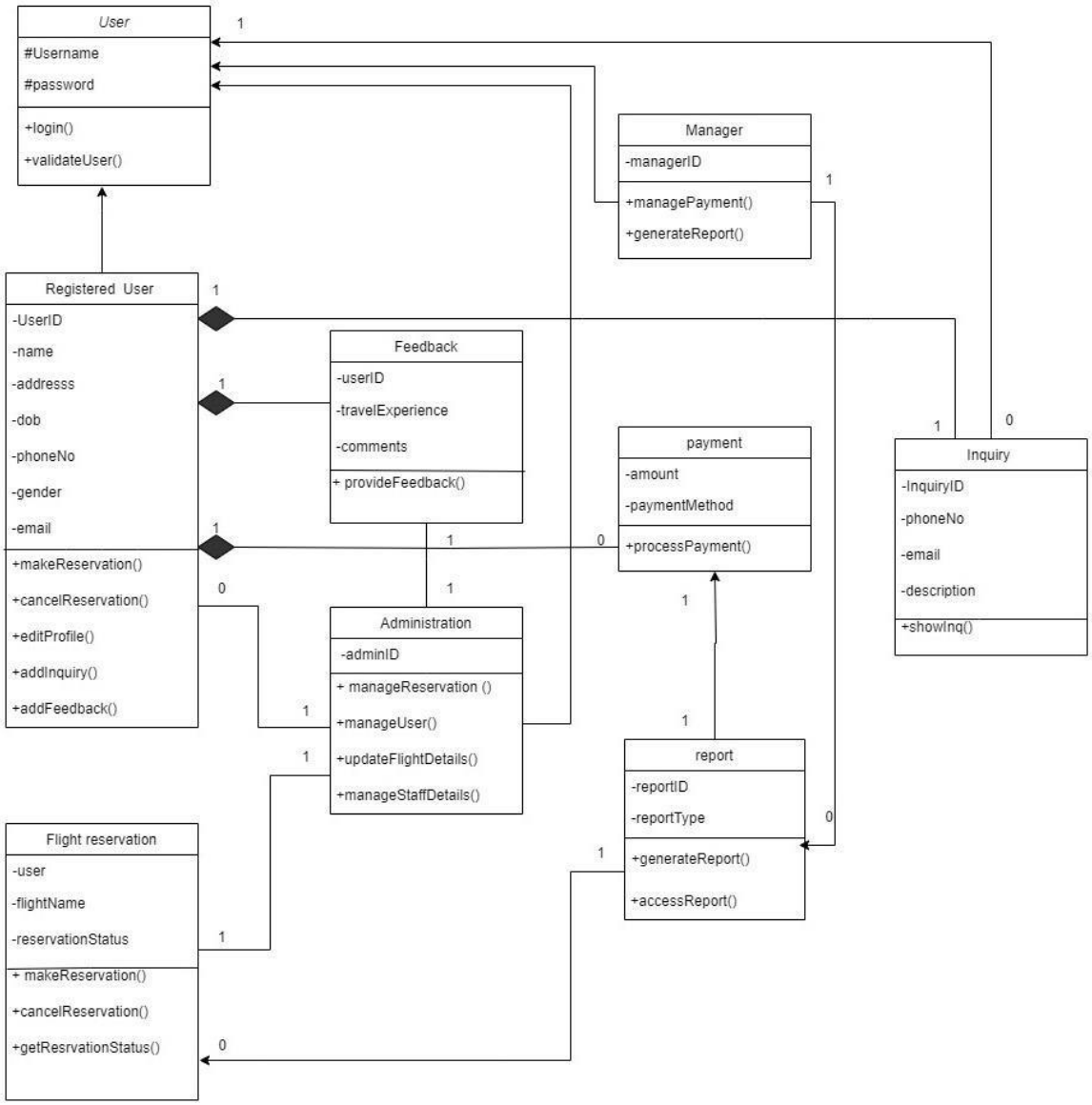
payment	
Responsibilities	collaborators
Manage account details	User
Process payments securely	
Handle transaction verification and confirmation	
Update ticket status	

Manager	
Responsibilities	Collaborators
Monitor the airline ticket reservation system	user
Analyze system performance	
Manage payment	payment
Generate report	report

Report	
Responsibilities	Collaborators
Generate report	
Access report	

Inquiry	
Responsibilities	Collaborators
Receive and address inquiries regarding reservation and system related	user
Provide response to user inquiry	admin

Class Diagram



Coding for the classes

```
#include<iostream>

#include<string>

using namespace std;



#pragma once

// Class - Administration


#include"User.h"

#include"FlightReservation.h"

#include"RegisteredUser.h"

#include"Feedback.h"

#define SIZE 10


class Administration : public User

{

private:

        int  AdminID;

        FlightReservation* flight[SIZE];

        RegisteredUser* regUserl[SIZE];

        Feedback* feedback[SIZE];


public:

        Administration();

        Administration(int adminid);

        void manageUser();
```

```
        void  updateFlightDetails();
        void  manageStaffDetails();
        void manageReservation();
        ~Administration();
};

#pragma once
//Class - User

class User
{
protected:
        char UserName[30];
        char Password[10];

public:
        User();
        User(const char username[], const char password[]);
        void displayUser();
        void login();
        void validateUser();
        ~User();

};

#include"FlightReservation.h"
#include"Payment.h"
```

```
#include"Manager.h"
#define SIZE 10



// report class
class Report
{
private:
	int ReportID;
	char ReportType[20];
	FlightReservation* flight[SIZE];
	Manager* manage;
	Payment* pay[SIZE];

public:
	Report();
	Report(int repoid, const char repotype[]);
	void generateReport();
	void accessReport();
	~Report();

};

#pragma once
//Class - RegisteredUser

#include"User.h"
```

```
#include"Feedback.h"
#include"Inquiry.h"
#include"Payment.h"
#define SIZE 10

class RegisteredUser : public User
{
private:
        char UserID[20];
        char Name[30];
        char Gender[10];
        char Address[40];
        char  DOB[20];
        char PhoneNo[10];
        char Email[30];
        Feedback* feed[SIZE];
        Inquiry* inq[SIZE];
        Payment* pay[SIZE];
        Administration* admin;

public:
        RegisteredUser();
        RegisteredUser(const char uid[], const char uname[], const char ugender[], const char
uadd[], const char udob[], const char uphone[], const char uemail[]);
        void  displayRUser();
        void makeReservation();
        void cancelReservation();
        void editProfile();
```

```
        void addInquiry();
        ~RegisteredUser();




};

#pragma once
//Class - Payment

#include"Manager.h"
class Payment
{
private:
        float Amount;
        char PaymentMethod[15];
        Manager* manage;

public:
        Payment();
        Payment(float amount, const char paymethod[]);
        void displayPayment();
        void processPayment();
        ~Payment();



};
```

```
#pragma once
//Class - Manager

#include"User.h"
#include"Report.h"
#include"Payment.h"
#define SIZE 10

class Manager : public User
{
private:
	int ManagerID;
	Report* rep[SIZE];
	Payment* pay[SIZE];

public:
	Manager();
	Manager(int manageid);
	void  managePayment();
	void generateReport();
	~Manager();


};

#pragma once
```

```
// Class - Inquiry
#include"User.h"

class Inquiry : public User
{
private:
	char InqID[10];
	char phoneNo[10];
	char Email[30];
	char Description[500];

public:
	Inquiry();
	Inquiry(const char inqid[], const char phoneno[], const char email[], const char descrip[]);
	void showInq();
	~Inquiry();


};

#pragma once
// Class - FlightReservation

#include "Administration.h"

class FlightReservation
{
```

```
private:
        char User[30];
        char FlightName[10];
        char ReservationStatus[10];
        Administration* admin;

public:
        FlightReservation();
        FlightReservation(const char user[], const char flight[], const char restatus[]);
        void displayFlightReservation();
        void makeReservation();
        void cancelReservation();
        void getReservationStatus();

};

#pragma once
//Class - Feedback

#include"Administration.h"
class Feedback
{
private:
        char UserId[10];
        char TravelExperience[500];
        char Comment[500];
        Administration* admin;
```

```
public:
	Feedback();
	Feedback(const char userid[], const char travelex[], const char comment[]);
	void provideFeedback();
	~Feedback();

};
```

```
//Administration.cpp

#include "Administration.h"
#include<iostream>
#include<cstring>
using namespace std;

Administration::Administration()
{
	AdminID = 0;
}

Administration::Administration(int adminid)
{
	AdminID = adminid;
```

```
}

void Administration::manageUser()
{

}

void Administration::updateFlightDetails()
{

}

void Administration::manageStaffDetails()
{

}

void Administration::manageReservation()
{

}

Administration::~Administration()
{

}
```

```
//Feedback.cpp


#include "Feedback.h"
#include<iostream>
#include<cstring>
using namespace std;


Feedback::Feedback()
{
	strcpy_s(UserId, "");
	strcpy_s(TravelExperience, "");
	strcpy_s(Comment, "");


}


Feedback::Feedback(const char userid[], const char travelex[], const char comment[])
{
	strcpy_s(UserId, userid);
	strcpy_s(TravelExperience, travelex);
	strcpy_s(Comment, comment);
}


void Feedback::provideFeedback()
{
	cout << "Feedback" << endl;
	cout << "Experience : " << TravelExperience << endl;
```

```
	cout << "Comment : " << Comment << endl;
	cout << endl;
}

Feedback::~Feedback()
{

}
```

```
//FlightReservation.cpp

#include "FlightReservation.h"
#include<iostream>
#include<cstring>
using namespace std;

FlightReservation::FlightReservation()
{
	strcpy_s(User, "");
	strcpy_s(FlightName, "");
	strcpy_s(ReservationStatus, "");

}
```

```
FlightReservation::FlightReservation(const char user[], const char flight[], const char restatus[])
{
	strcpy_s(User, user);
	strcpy_s(FlightName, flight);
	strcpy_s(ReservationStatus, restatus);

}

void FlightReservation::displayFlightReservation()
{
	cout << "Flight Reservation Details" << endl;
	cout << "User : " << User << endl;
	cout << "Flight Name : " << FlightName << endl;
	cout << endl;
}

void FlightReservation::makeReservation()
{

}

void FlightReservation::cancelReservation()
{

}
```

```
void FlightReservation::getReservationStatus()
{

}

FlightReservation::~FlightReservation()
{

}
```

```
//Inquiry.cpp

#include "Inquiry.h"
#include<iostream>
#include<cstring>
using namespace std;

Inquiry::Inquiry()
{
	strcpy_s(InqID, "");
	strcpy_s(phoneNo, "");
```

```
	strcpy_s(Email, "");
	strcpy_s(Description, "");

}

Inquiry::Inquiry(const char inqid[], const char phoneno[], const char email[], const char descrip[])
{
	strcpy_s(InqID, inqid);
	strcpy_s(phoneNo, phoneno);
	strcpy_s(Email, email);
	strcpy_s(Description, descrip);
}

void Inquiry::showInq()
{
	cout << "Inquiry" << endl;
	cout << InqID << ":" << Description << endl;
	cout << endl;
}

Inquiry::~Inquiry()
{

}

//Manager.cpp
```

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

Manager::Manager()
{
	ManagerID = 0;
}

Manager::Manager(int manageid)
{
	ManagerID = manageid;
}

void Manager::managePayment()
{

}

void Manager::generateReport()
{

}

Manager::~Manager()
{
```

```
}



//Payment.cpp

#include "Payment.h"
#include<iostream>
#include<cstring>
using namespace std;

Payment::Payment()
{
	Amount = 0.0;
	strcpy_s(PaymentMethod, "");

}

Payment::Payment(float amount, const char paymethod[])
{
	Amount = amount;
	strcpy_s(PaymentMethod, paymethod);
}
void Payment::displayPayment()
{
	cout << "Payment" << endl;
```

```
        cout << "Method : " << endl;
        cout << "Total Amount : " << Amount << endl;
        cout << endl;

}

void Payment::processPayment()
{

}

Payment::~Payment()
{

}



//RegisteredUser.cpp

#include "RegisteredUser.h"

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

RegisteredUser::RegisteredUser()
{
```

```
	strcpy_s(UserID, "");
	strcpy_s(Name, "");
	strcpy_s(Gender, "");
	strcpy_s(Address, "");
	strcpy_s(DOB, "");
	strcpy_s(PhoneNo, "");
	strcpy_s(Email, "");
}

RegisteredUser::RegisteredUser(const char uid[], const char uname[], const char ugender[], const char uadd[], const char udob[], const char uphone[], const char uemail[])
{
	strcpy_s(UserID, uid);
	strcpy_s(Name, uname);
	strcpy_s(Gender, ugender);
	strcpy_s(Address, uadd);
	strcpy_s(DOB, udob);
	strcpy_s(PhoneNo, uphone);
	strcpy_s(Email, uemail);

}

void RegisteredUser::displayRUser()
{
	cout << "Registered User Details" << endl;
	cout << "Registered User ID : " << UserID << endl;
	cout << "Name : " << Name << endl;
	cout << "Gender : " << Gender << endl;
```

```
        cout << "DOB : " << DOB << endl;
        cout << "Phone No : " << PhoneNo << endl;
        cout << "Email : " << Email << endl;
        cout << endl;
}

void RegisteredUser::makeReservation()
{

}

void RegisteredUser::cancelReservation()
{

}

void  RegisteredUser::editProfile()
{

}

void RegisteredUser::addInquiry()
{

}

RegisteredUser::~RegisteredUser()
```

```
{


}

//Report.cpp

#include "Report.h"
#include<iostream>
#include<cstring>
using namespace std;

Report::Report()
{
	ReportID = 0;
	strcpy_s(ReportType, "");


}

Report::Report(int repoid, const char repotype[])
{
	ReportID = repoid;
	strcpy_s(ReportType, repotype);
}

void Report::generateReport()
{
```

```
}

void Report::accessReport()
{

}

Report::~Report()
{

}



//User.cpp
#include "User.h"
#include<iostream>
#include<cstring>
using namespace std;

User::User()
{
	strcpy_s(UserName, "");
	strcpy_s(Password, "");

}

User::User(const char username[], const char password[])
```

```
{
	strcpy_s(UserName, username);
	strcpy_s(Password, password);
}

void User::displayUser()
{
	cout << "User Name : " << UserName << endl;
	cout << endl;
}

void User::login()
{

}

void User::validateUser()
{

}

User :: ~User()
{

}
```

```
// airline.cpp : This file contains the 'main' function. Program execution begins and ends there.
//Main.cpp

#include"User.h"
#include"RegisteredUser.h"
#include"Administration.h"
#include"Feedback.h"
#include"FlightReservation.h"
#include"Inquiry.h"
#include"Manager.h"
#include"Payment.h"
#include"Report.h"

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

int main(void)
{
	//Data insert RegisteredUser
	RegisteredUser* regUser = new RegisteredUser("U001", "Kamal Samantha", "Male", "Havelock City, Colombo 05, Sri Lanka.", "1984-07-06", "0713245631", "kamals@gmail.com");
	regUser->displayRUser();
```

```
        //Data insert Inquiry
        Inquiry* inq = new Inquiry("Inq10", "0771534523", "malki1@gmail.com", "Visa Requirement
Inquiry");
        //Display Inquiry Details
        inq->showInq();



        //data insert FlightReservation
        FlightReservation* reserve = new FlightReservation("Nimal Perera", "EK135", "Completed");
        //Display FlightReservation Details
        reserve->displayFlightReservation();



        //data insert Administration
        Administration* admin = new Administration(10);



        //data insert Report
        Report* repo = new Report(30, "Baggage Allowance");



        //data insert Payment
        Payment* pay = new Payment(150000.00, "Credit Card");
        pay->displayPayment();



        //data insert Manager
```

```
        Manager* manage = new Manager(26);



        //data insert Feedback
        Feedback* feedb = new Feedback("U002", "Good", "Customer Service is very good.");
        feedb->provideFeedback();



        delete regUser;
        delete inq;
        delete reserve;
        delete admin;
        delete repo;
        delete manage;
        delete pay;
        delete feedb;
}
```

Class user- Jayanath D.T

Class registered user- Jayanath D.T

Class flight reservation- Sharvajen.S

Class feedback- Sharvajen.S

Class administration- Sharvajen.S

.TClass manager- Harikirushna.T

Class payment- Vijayananthan.D

Class report- Vijayananthan.D

Class inquiry- Pakeershan.C