

Sri Lanka Institute of Information Technology



Topic : Airline Ticket Reservation System

Group no : MLB_WD_03.02_04

Campus : Malabe

Submission Date : 20 / 05 / 2022

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
IT21249266	Peiris A. R. D.	0777562621
IT21479250	Gunathilaka I.U.	0788915317
IT21487934	Chamoda G.M.W.Y.	0706060370
IT21470004	Bopitiya S. R.	0768525475
IT21562488	Anjana A. H. S.	0702171249

Noun Verb Analysis

System Requirements

- Passenger can register to the system by giving their details such as name, address, DOB, email address
- Passenger can view and update user profile
- Passenger can view flight details, check flight availability, and view ticket pricing
- Passenger can select meal type and seat type
- Passenger can reserve flights using payment portal
- Passenger can do the payment via PayPal, Debit Card/Credit Card and passenger can confirm the payment
- System validates payment and updates number of seats available in flight. Then system sends E-receipt to customer's email address
- The people who reserved tickets have the option of cancelling their reservation.
- Customer can request help and support from customer service agent and provide feedback.
- Customer service agent can directly contact other Finance, Admin, HR, Booking departments
- Customer service agent can access passenger details
- Airline Reservation Agent can issue flight tickets online via email or issue at the counter to the passenger once confirming the payment and other documents.
- System should allow the Airline Reservation Agent access to view flight details, flight reservations, customer payments by the passenger/customer.
- System should allow the Airline Reservation Agent to update passenger details in the system database.
- System should allow access for the Airline Reservation Agents to view and validate customer payments.
- System should allow airport manager to view the website as Staff, Flight Details, Payment Details, Passenger details
- System should allow timely upgrades on server management
- System should allow access to Airport Manager to adjust the system such as flight details, payment details, staff details.
- Airport Managers can interact with the database and guide the administrators.
- System should allow the administrator to access and view the flight, reservation and payment details of customer
- Administrator can validate reservations and report to managers.
- Admin can also update the database and handle the payment gateway.
- Admin can back up the system database

Nouns

Passenger

System

Details

Name

Address

DoB

Email address

User Profile

Flight details

Flight availability

Ticket Pricing

Meal Type

Seat Type

Flights

Payment Portal

Payment

PayPal

Debit Card

Credit Card

Number of seats

E-receipt

Customer's email address

People

Tickets

Reservation

Customer

Help and support

Customer service agent

Feedback

Finance

Admin

HR

Booking departments

Passenger details

Airline Reservation Agent

Flight tickets

E-mail

Counter

Documents

Flight reservation

Customer payment

System database

Airport manager

Website

Staff

Payment details

Upgrades

Server management

Database

Administrator

Managers

Payment gateway

Admin

Classes

Passenger

System

Flight

Payment

Tickets

Reservation

Staff

Redundant

Details

User Profile

Flight availability

Customer's email address

Customer

Passenger details

Flight tickets

E-mail Address

Flight reservation

Customer payment

Payment details

Managers

Admin

PayPal

Debit Card

Credit Card

Payment gateway

Event/Operation

Using payment Portal

Send E-receipt

Request Help and support

Provide feedback

Out of scope

Finance

Admin

HR

Booking departments

Counter

Documents

Website

Upgrades

Server management

Customer service agent

Airline Reservation Agent

Airport manager

Administrator

System database

Meta Language

People

Attributes

Name

Address

DOB

Email

Flight details

Ticket Pricing

Meal Type

Seat Type

Number of seats

Verbs

Register

Give Passenger Details

View User profile

Update User profile

View flight details

Check flight availability

View ticket pricing

Select meal type

Select seat type

Reserve flights

Confirm payment

Validate payment

Update number of seats

Cancelling reservation

Request help and support

Provide feedback

Contact other departments

Access passenger details

Issue ticket details online

Issue ticket details physically

Allow access to airline reservation agent

View flight details

Update passenger details

View customer payments

Validate customer payments

Allow access for airport manager to website
Allow timely upgrades
Adjust system
Interact with database
Guide the administrators
Allow administrator access
View flight, reservation and payment details of customer
Validate reservations
Generate reports
Update database
Handle payment gateway
Backup system database

Passenger

Register
Give Passenger Details
View User profile
Update User profile
View ticket pricing
Request help and support
Provide feedback

Flight

View flight Details
Check flight availability

Payment

Confirm payment
Validate payment

Tickets

Issue ticket details online

Issue ticket details physically

Reservation

Select meal type

Select seat type

Reserve flights

Cancelling reservation

Validate reservations

Generate reports

Staff

Contact other department

Update passenger details

View customer payments

Interact with database

Update database

View reservation details of customer

CRC Cards

Passenger	
Responsibilities	Collaborators
Register to system	
Give Passenger Details	
Book tickets	Ticket
Do payment	Payment
Request help and support	Staff

Flight	
Responsibilities	Collaborators
View flight Details	
Check flight availability	Passenger
Validate ticket	Staff, Ticket

Payment	
Responsibilities	Collaborators
Confirm payment	Passenger
Validate payment	Staff

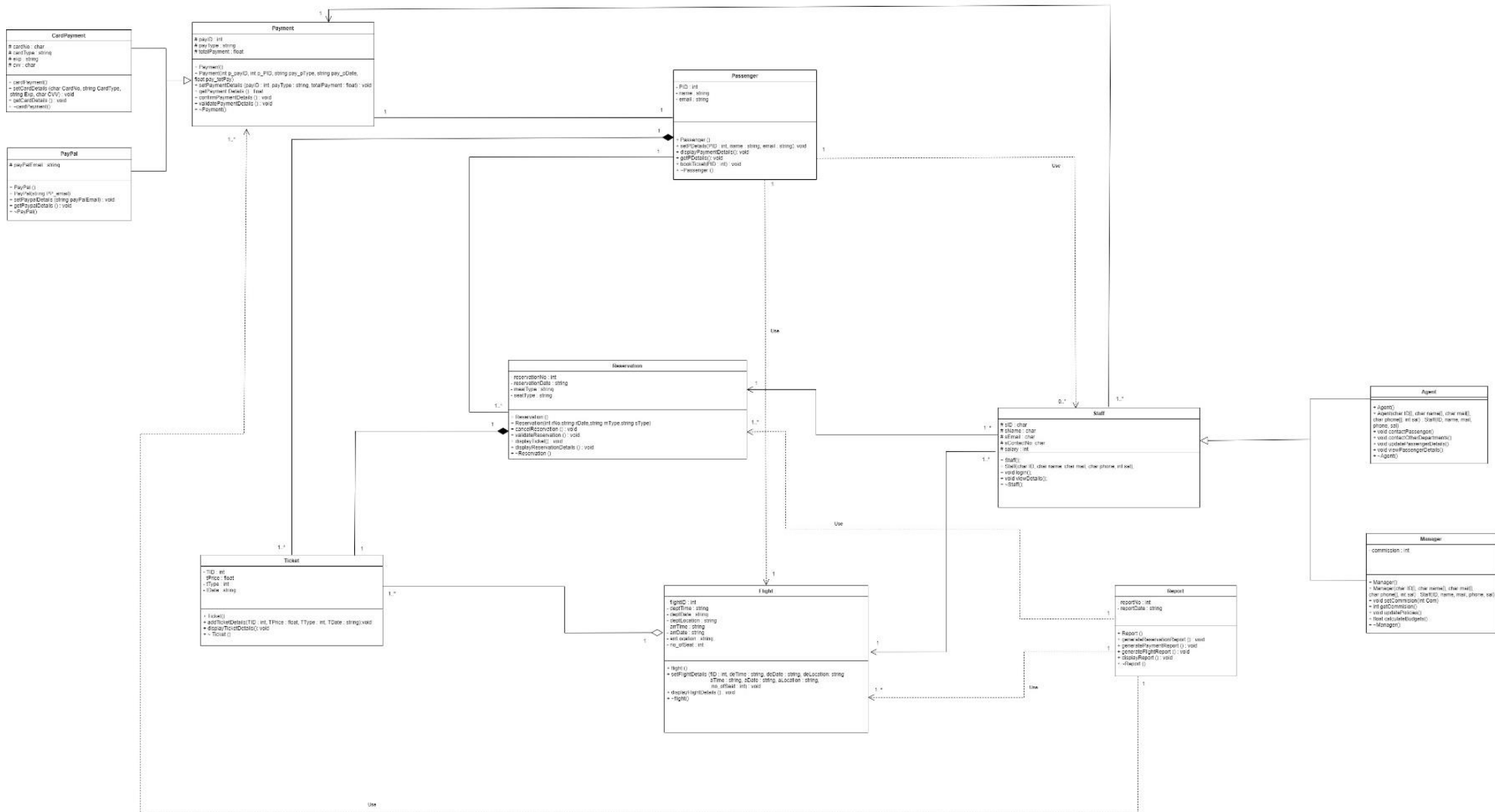
Ticket	
Responsibilities	Collaborators
Issue ticket	Passenger
Add ticket details	
Delete ticket details	

Reservations	
Responsibilities	Collaborators
Add reservation details	Passenger
Reserve flights	
Cancelling reservation	Passenger
Validate reservations	Staff, Ticket

Staff	
Responsibilities	Collaborators
Update passenger details	Passenger
View passenger payments	Payment
Update database	

Report	
Responsibilities	Collaborators
Generate Reservation reports	Reservation
Generate Payment reports	Payment
Flight reports	Flight

Class Diagram



Coding

Agent.h

//Created by IT21249266 PEIRIS ARD

#pragma once

#include "Staff.h"

class Agent : public Staff

{

public:

 Agent();

 Agent(char ID[], char name[], char mail[], char phone[], int sal)

 : Staff(ID, name, mail, phone, sal);

 void contactPassenger();

 void contactOtherDepartments();

 void updatePassengerDetails();

 void viewPassengerDetails();

 ~Agent();

};

Agent.cpp

//Created by IT21249266 PEIRIS ARD

#include "Agent.h"

#include "Staff.h"

#include <cstring>

using namespace std;

Agent::Agent() {

```
}
```

```
Agent::Agent(char ID[], char name[], char mail[], char phone[], int sal)
```

```
    : Staff(ID, name, mail, phone, sal) {
```

```
}
```

```
void Agent::contactPassenger() {
```

```
}
```

```
void Agent::contactOtherDepartments() {
```

```
}
```

```
void Agent::updatePassengerDetails() {
```

```
}
```

```
void Agent::viewPassengerDetails() {
```

```
}
```

```
Agent::~~Agent() {
```

```
}
```

CardPayment.h

```
//Created by IT21470004 - S. R. Bopitiya
```

```
#pragma once
```

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

```
#include <string>
```

```
using namespace std;
```

```
class CardPayment : public Payment{ //inheritance relationship
protected:
    char cardNo [20];
    string cardType;
    string exp;
    char cvv [3];

public:
    cardPayment();
    void setCardDetails(char CardNo[20], string CardType, string Exp, char CVV[3]);
    void getCardDetails();
    ~cardPayment();
};
```

CardPayment.cpp

//Created by IT21470004 - S. R. Bopitiya

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

CardPayment::CardPayment()
{
    strcpy(cardNo, "0000 0000 0000 0000");
    cardType = "";
    exp = "MM/YY";
    strcpy(cvv, "000");
}
```

```
void CardPayment::setCardDetails(char CardNo[20], string CardType, string Exp, char CVV[3])
```

```
{  
    strcpy(cardNo, CardNo);  
    cardType = CardType;  
    exp = Exp;  
    strcpy(cvv, CVV);  
}
```

```
void CardPayment::getCardDetails()
```

```
{  
    getPaymentDetails();  
    cout << "Card No.: " << cardNo << endl;  
    cout << "Card Type: " << cardType << endl;  
    cout << "Card Expiration Date: " << exp << endl;  
    cout << "Card CVV: " << cvv << endl;  
}
```

```
CardPayment::~~CardPayment()
```

```
{  
    cout << "Deleting Card Payment Details" << endl;  
}
```

Flight.h

```
// created by IT21487934-Chamoda G.M.W.Y.
```

```
#pragma once
```

```
#include "Staff.h"
```

```
#include "Passenger.h">
```



```
#include "Ticket.h"
```

```
#include "Report.h"
```

```
#include <string>
```

```
class Flight {
```

```
private:
```

```
    int flightID;
```

```
    string deptTime;
```

```
    string deptDate;
```

```
    string deptLocation;
```

```
    string arrTime;
```

```
    string arrDate;
```

```
    string arrLocation;
```

```
    int noOfSeats;
```

```
    ticket *t[2];
```

```
public:
```

```
    Flight();//constructor
```

```
    void setflightDetails(int fID , string deTime, string deDate, string deLocation, string  
aTime, string aDate, string aLocation, int no_ofSeats);
```

```
    void displayflightDetails();
```

```
    void addTicketDetails(ticket *t1,ticket *t2)
```

```
    ~Flight();//destructure
```

```
};
```

Flight.cpp

```
//IT21487934 Chamoda G.M.W.Y.
```

```
#include <iostream>
```

```
#include "Flight.h"
```

```
#include <string>
```

```
using namespace std;
```

```
Flight::Flight()
```

```
{
```

```
    flightID=0;
```

```
    deptTime="";
```

```
    deptDate="DD/MM/YYYY";
```

```
    deptLocation="";
```

```
    arrTime="";
```

```
    arrDate="";
```

```
    arrLocation="";
```

```
    noOfSeats=0;
```

```
}
```

```
void addTicketDetails(ticket *t1,ticket *t2)
```

```
{
```

```
}
```

```
void Flight::setFlightDetails(int fID , string deTime, string deDate, string deLocation, string  
aTime, string aDate, string aLocation, int no_ofSeats)
```

```
{
```

```
    flightID=fID;
```

```
    deptTime=deTime;
```

```
    deptDate=deDate;
```

```
    deptLocation=deLocation;
```

```
    arrTime=aTime;
```

```
    arrDate=aDate;
```

```
        arrLocation=aLocation;
        noOfSeats=no_ofSeats;

    }

void Flight::displayFlightDetails()
{

}

Flight::~Flight()
{
    cout<<"flight shutting down"<<endl;
}
```

Manager.h

//Created by IT21249266 PEIRIS ARD

```
#include "Staff.h"
#pragma once
class Manager : public Staff
{
private:
    int commission;

public:
    Manager();
    Manager(char ID[], char name[], char mail[], char phone[], int sal)
        : Staff(ID, name, mail, phone, sal);
```

```
        void setCommision(int Com);  
        int getCommision();  
        void updatePolicies();  
        float calculateBudgets();  
        ~Manager();  
};
```

Manager.cpp

//Created by IT21249266 PEIRIS ARD

```
#include "Manager.h"  
#include "Staff.h"  
#include <cstring>  
using namespace std;  
  
Manager::Manager() {  
  
    commission = 0;  
}  
  
Manager::Manager(char ID[], char name[], char mail[], char phone[], int sal)  
    : Staff(ID, name, mail, phone, sal) {  
  
}  
  
void Manager::setCommision(int Com) {  
    commission = Com;  
}  
  
int Manager::getCommision() {  
    return commission;  
}
```

```
void Manager::updatePolicies() {  
  
}  
  
float Manager::calculateBudgets() {  
  
}  
  
Manager::~~Manager() {  
  
}
```

Passeneger.h

//Created by IT21479250 Gunathilaka I.U.

```
#define SIZE 3
```

```
class Passenger{  
    private:  
        int PID;  
        string name;  
        string email;  
        Payment *paymt;  
        Ticket *tikt[SIZE];  
    public:  
        Passenger();  
        void setDetails(int pPID, string pname, string pemail);  
        void displayPaymentDetails(Payment *pPaymt);  
        void getPDDetails();  
        void bookTicket();  
        ~Passenger();  
};
```

Passenegr.cpp

//Create by IT21479250 Gunathilaka I.U.

#include "Passenger.h"

#include <iostream>

using namespace std;

Passenger::Passenger(){

 PID = 0000;

 name = "";

 email = "";

}

void Passenger::setDetails(int pPID, string pname, string pemail){

 PID = pPID;

 name = pname;

 email = pemail;

}

void Passenger::displayPaymentDetails(Payment *pPaymt){

 cout << "Passenger ID : " << PID << endl;

 paymt = pPaymt;

 cout << "The payment details : " << paymt->getPaymentDetails();

}

void Passenger::getPDDetails(){

 cout << "Passenger ID : " << pID << endl;

 cout << "Passenger name : " << name << endl;

 cout << "Passenger email : " << email << endl;

}

void Passenger::bookTicket(){

 for (int i = 0; i < SIZE; i++)

 tikt[i] = new Ticket();

}

```
Passenger::~~Passenger(){
    for (int i = 0; i < SIZE; i++)
        delete tikt[i];
}
```

PayPal.h

//Created by IT21470004 - S. R. Bopitiya

```
#pragma once
```

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

```
#include <string>
```

```
class PayPal : public Payment{
    protected:
        string payPalEmail;

    public:
        PayPal();
        PayPal(string PP_email);
        void setPaypalDetails(string payPalEmail);
        void getPaypalDetails();
        ~PayPal();
};
```

PayPal.cpp

//Created by IT21470004 - S. R. Bopitiya

```
#include <iostream>
```

```
#include <string>
```

```
#include "PayPal.h"
```

```
using namespace std;
```

```
PayPal::PayPal()
```

```
{  
    payPalEmail = "";  
}
```

```
PayPal::PayPal(string PP_email)
```

```
{  
    payPalEmail = PP_email;  
}
```

```
void PayPal::setPaypalDetails(string payPalEmail)
```

```
{  
  
}
```

```
void PayPal::getPaypalDetails()
```

```
{  
  
}
```

```
PayPal::~~PayPal()
```

```
{  
    cout << "Deleting PayPal Details: " << endl;  
}
```

Payment.h

```
//Created by IT21470004 - S. R. Bopitiya
```

```
#pragma once
```



```
#include "Passenger.h"

#include "Staff.h"

#include "Report.h"

#include <string>

using namespace std;

class Payment
{
protected:
    int payID;
    string payType;
    float totalPayment;
    Passenger* p; //passenger dynamic object

public:
    Payment(); //constructor
    Payment(int p_payID; int p_PID, string pay_pType, string pay_pDate, float pay_totPay);
    void setPaymentDetails(int payID, string payType, float totalPayment);
    float getPaymentDetails();
    void confirmPaymentDetails();
    void validatePaymentDetails();
    ~Payment(); //destructor
}
```

Payment.cpp

//Created by IT21470004 - S. R. Bopitiya

```
#include <iostream>

#include "Payment.h"

using namespace std;
```

```
Payment::Payment()
```

```
{  
    payID = "";  
    payType = "";  
    totalPayment = 0;  
}
```

```
Payment::Payment(int p_payID, int p_PID, string pay_pType, string pay_pDate, float  
pay_totPay)
```

```
{  
    payID = p_payID;  
    passengerID = p_PID;  
    payType = pay_pType;  
    payDate = pay_pDate;  
    totalPayment = pay_totPay;  
}
```

```
void Payment::setPaymentDetails(int payID, string payType, float totalPayment)
```

```
{  
  
}
```

```
void Payment::getPaymentDetails()
```

```
{  
    cout << "Payment ID: " << payID << endl;  
    cout << "Payment Type: " << payType << endl;  
    cout << "Payment Amount: " << TotalPayment << endl;  
}
```

```
void Payment::confirmPaymentDetails()
```

```
{  
  
}  
  
void Payment::validatePaymentDetails()  
{  
  
}  
  
Payment::~~Payment()  
{  
    cout << "Deleting Payments" << endl;  
}
```

Report.h

//Created by IT21470004 - S.R. Bopitiya

```
#pragma once  
#include "Reservation.h"  
#include "Flight.h"  
#include "Payment.h"  
#include <string>  
using namespace std;  
  
class Report  
{  
    private:  
        int reportNo;  
        string reportDate;
```

```
public:
    Report(); //constructor
    void generateReservationReport(Reservation* r);
    void generatePaymentReport(Payment* p);
    void generateFlightReport(Flight* f);
    void displayReport();
    ~Report(); //destructor
};
```

Report.cpp

//Created by IT21470004 - S.R. Bopitiya

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

```
Report::Report()
```

```
{
    reportNo = 0;
    reportDate = "DD/MM/YYYY";
}
```

```
void Report::generateReservationReport(Reservation* r)
```

```
{

}
```

```
void Report::generateFlightReport(Flight* f)
```

```
{
```

```

    }

    void Report::generatePaymentReport(Payment* p)
    {

    }

    Report::~~Report()
    {
        cout << "Deleting Reports" << endl;
    }

```

Reservation.h

//created by IT21562488 - A.H.S ANJANA

```

#pragma once

#include "Ticect.h"
#include "Passenger.h"
#include "Staff.h"
#include "Report.h"
#include<iostream>
#include <string>
using namespace std;

class Reservation
{
private:
    int reservationNo;
    string reservationDate;
    string mealType;
    string seatType;

```

```
Ticket *tic;

passenger *pas;

public:
    Reservation()
    Reservation(int rNo,string rDate,string mType,string sType)
    void addTicket(Ticket *t)
    void addPassenger(Passenger *p)
    void cancelReservation ()
    void validateReservation ()
    void displayTicket ()
    void displayReservationDetails ()
    ~Reservation ()
};
```

Reservation.cpp

```
//created by IT21562488 - A.H.S ANJANA
```

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

```
Reservation::Reservation()
{
    reservationNo = 00;
    reservationDate = "DD/MM/YYYY";
    mealType = " ";
```

```
        seatType = " ";

    }

Reservation::Reservation(int rNo,string rDate,string mType,string sType)
{

    reservationNo = rNo;
    reservationDate = rDate;
    mealType = mType;
    seatType = sType;

}

void Reservation::addTicket(Ticket *t)
{

}

void Reservation::addPassenger(Passenger *p)
{

}

void Reservation::cancelReservation ()
{

}

void Reservation::validateReservation ()
{

}
```

```
void Reservation::displayReservationDetails ()
{

}

void Reservation::displayTicket ()
{

    tic -> displayTicketDetails();
}

Reservation::~Reservation ()
{

    cout<<"Reservation shutting down"<<endl;
    delete tic;
}
```

Staff.h

//Created by IT21249266 PEIRIS ARD

```
#pragma once
//parent class
class Staff
{
protected:
    char sID[10];
    char sName[20];
    char sEmail[100];
    char sContactNo[10];
    int salary;
```



```
public:
    Staff();
    Staff(char ID[], char name[], char mail[], char phone[], int sal);
    void login();
    void viewDetails();
    ~Staff();
};
```

Staff.cpp

```
//Created by IT21249266 PEIRIS ARD
```

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

```
Staff::Staff() {

    strcpy(sID, "");
    strcpy(sName, "");
    strcpy(sEmail, "");
    strcpy(sContactNo, "");
    salary = 0;
}
```

```
Staff::Staff(char ID[], char name[], char mail[], char phone[], int sal) {

    strcpy(sID, ID);
    strcpy(sName, name);
    strcpy(sEmail, mail);
    strcpy(sContactNo, phone);
```

```
        salary = sal;
    }

    void Staff::login() {

    }

    void Staff::viewDetails() {

    }

    Staff::~~Staff() {

    }
```

Ticket.h

//Created by IT21479250 Gunathilaka I.U.

```
class Ticket{
    private:
        int TID;
        float tPrice;
        int tType;
        string tDate;
    public:
        Ticket();
        void addTicketDetails (int pTID, float ptPrice, int ptType, string ptDate);
        void displayTicketDetails();
        ~Ticket();

};
```

Ticket.cpp

//Created by IT21479250 Gunathilaka I.U.

```
#include "Ticket.h"
```

```
#include <iostream>
```

```
using namespace std;
```

```
Ticket::Ticket(){
```

```
    TID = 0;
```

```
    tPrice = 0.00;
```

```
    tType = 0;
```

```
    tDate = "0000/00/00";
```

```
}
```

```
void Ticket::addTicketDetails (int pTID, float ptPrice, int ptType, string ptDate){
```

```
    TID = pTID;
```

```
    tPrice = ptPrice;
```

```
    tType = ptType;
```

```
    tDate = ptDate;
```

```
}
```

```
void Ticket::displayTicketDetails(){
```

```
    cout << "Ticket ID : " << TID << endl;
```

```
    cout << "Ticket price : " << tPrice << endl;
```

```
    cout << "Ticket Type : " << tType << endl;
```

```
    cout << "Ticket issued date : " << tDate << endl;
```

```
}
```

```
Ticket::~~Ticket(){
```

```
    cout << "Ticket Deleted" << endl;
```

```
}
```

Main.cpp

```
#include <iostream>
```

```
#include <string>
```

```
#include "Passenger.h"
```

```
#include "Flight.h"
```

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

```
#include "PayPal.h"
```

```
#include "CardPayment.h"
```

```
#include "Staff.h"
```

```
#include "Agent.h"
```

```
#include "Manager.h"
```

```
#include "Reservation.h"
```

```
#include "Ticket.h"
```

```
#include "Report.h"
```

```
using namespace std;
```

```
int main(){
```

```
    // Creating objects
```

```
    Passenger* p = new Passenger();
```

```
    Flight* f = new Flight();
```

```
    Payment* pay = new Payment();
```

```
    PayPal* pp = new PayPal();
```

```
    CardPayment* c = new CardPayment();
```

```
    Staff* s = new Staff();
```

```
    Agent* a = new Agent();
```

```
    Manager* m = new Manager();
```

```
    Reservation* r = new Reservation();
```

```
    Ticket* t = new Ticket();
```

```
Report* rpvt = new Report();
```

```
//Deleting objects
```

```
delete p;
```

```
delete f;
```

```
delete pay;
```

```
delete pp;
```

```
delete c;
```

```
delete s;
```

```
delete a;
```

```
delete m;
```

```
delete r;
```

```
delete t;
```

```
delete rpvt;
```

```
return 0;
```

```
}
```

Individual Contribution

Student ID	Student Name	Contribution
IT21249266	Peiris A. R. D.	Staff, Manager, Agent
IT21479250	Gunathilaka I.U.	Passenger, Ticket
IT21487934	Chamoda G.M.W.Y.	Flight
IT21470004	Bopitiya S. R.	Report, Payment, CardPayment, PayPal
IT21562488	Anjana A. H. S.	Reservation