

Topic: Airline Ticket Reservation Sysytem

Group no: MLB\_02.01\_12

Campus: Malabe

Submission Date : 20th May

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** | **Contributions** |
| IT21202018 | Charindu Ranasinghe | 0786325662 | Customer UML and Customer Class coding |
| IT21199080 | R.M.Harendra Nayanamudu | 0703844385 | Flights UML and Flights Class coding |
| IT21199394 | Kalana Lasith Dewapriya | 0765269178 | Payment and Vaccination UML and Payment and Vaccination Class coding |
| IT21204302 | Sajeevan S | 0778131381 | Ticket and Visa UML and Ticket And Visa Class coding |
| IT21200106 | Kasthuriarachchi K.A.S. N | 0778357821 | Passport and Class Type UML and Passport and Class Type coding |

# Scenario

1. A customer can visit the airline ticket reservation website using the URL and browse through the website
2. Customer needs to register to the system using details such as name, address, NIC.
3. Once registered ,customers can add payment methods, visa and passport details and vaccination details.
4. Customer can search for flights by destination/date.
5. Customer can choose which class type to travel in each flight.
6. A total amount is displayed for booked tickets and customer can choose a payment method provided or add a new payment method.
7. Customer confirms the reservation, the ticket is reserved and marked in the system.
8. The system updates the remaining tickets for flights after a reservation is confirmed.
9. The Airline Staff validates payments.
10. The Airline Staff sends an email to the customer confirming the payment with a receipt.
11. Customers can leave feedback after their reserved flights are travelled.
12. The Administrator can review feedbacks sent by customers
13. Administrator can add new flights, change schedules, and remove flights.

# Noun/Verbs

1. A customer can visit the airline ticket reservation website using the URL and browse through the website
2. Customer needs to register to the system using details such as name, address, NIC.
3. Once registered ,customers can add payment methods, visa and passport details and vaccination details.
4. Customer can search for flights by destination/date.
5. Customer can choose which Class Type to travel in each flight.
6. A total amount is displayed for booked tickets and customer can choose a payment method provided or add a new payment method.(Debit card, Credit card, Paypal)
7. Customer confirms the reservation, the ticket is reserved and marked in the system.
8. The system updates the remaining tickets for flights after a reservation is confirmed.
9. The Airline Staff validates payments.
10. The Airline Staff sends an email to the customer confirming the payment with a receipt.
11. Customers can leave feedback after their reserved flights are travelled.
12. The Administrator can review feedbacks sent by customers
13. Administrator can add new flights, change schedules, and remove flights.

# Identified Nouns

Customer

System

Ticket

Name,address,NIC

Visa

Passport

Payment

Vaccination

Flights

Destination/Date

Class Type

Debit card, Credit card, Paypal

Airline Staff

Administrator

Email

# Identify Classes using noun/verb Analysis

Customer - Class

System – Class

Ticket - Class

Name,address,NIC - Attribute

Visa - Class

Passport - Class

Payment -Class

Vaccination - Class

Flights -Class

Destination/Date -Attributes

Class Type -Class

Debit card, Credit card, Paypal - Attributes

Airline Staff -Class

Administrator -Class

Email -Attribute

# Identified Classes

Customer – Class

Tickets - Class

Visa - Class

Passport - Class

Payment -Class

Vaccination - Class

Flights -Class

Class Type -Class

Airline Staff -Class

# CRC Cards

|  |  |
| --- | --- |
| Customer | |
| Responsibilities | Collaborators |
| Register to the airline ticket reservation system | Passport |
|  | Visa |
|  | Ticket |

|  |  |
| --- | --- |
| Tickets | |
| Responsibilities | Collaborators |
| Reserve seat in flight | Customer |
| Update tickets | Class Type |
|  | Flights |
|  | Payment |

|  |  |
| --- | --- |
| Visa | |
| Responsibilities | Collaborators |
| Confirm visa documents | Customer |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Passport | |
| Responsibilities | Collaborators |
| Add Passport Details | Customer |
| Validate Passport Details |  |

|  |  |
| --- | --- |
| Payment | |
| Responsibilities | Collaborators |
| Choose Payment Method | Tickets |
| Add payment |  |
| Update Payment |  |
| Validate Payment |  |

|  |  |
| --- | --- |
| Vaccination | |
| Responsibilities | Collaborators |
| Add Vaccination Details |  |
| Validate Vaccination Details |  |

|  |  |
| --- | --- |
| Flights | |
| Responsibilities | Collaborators |
| Search Flights | Tickets |

|  |  |
| --- | --- |
| Class Type | |
| Responsibilities | Collaborators |
| Choose Class type | Tickets |

# Class Diagram For Airline Ticket Reservation System

Diagram, timeline

Description automatically generated

# Coding

#include <iostream>

#include <string>

using namespace std;

//classes

class Customer;

class Tickets;

class Visa;

class Passport;

class Payment;

class Vaccination;

class Flights;

class Class\_type;

class customer {

private:

char customerName[], char customerID[], char nic[];

double contactNumber;

public:

Customer();

Customer(char cName[], char cID[], char nic[], double cNo);

char getCustomerName();

char getNIC();

char getContactNo();

void displayCustomerDetails();

~Customer();

};

class Passport

{

private:

char passportId[10], passportIssueDate[10], passportExpDate[10];

public:

Passport();

Passport(char passid[], char passIssDate[], char passExpDate[]);

void displayPassportDetails();

~Passport();

};

class Tickets

{

private:

char ticketId[10], issuedDate[10], expiredDate[10], classType[10];

public:

Tickets();

Tickets(char tId[], char issDate[], char expDate[], char classTy[]);

void addReservation();

void displayTicketDetails();

~Tickets();

};

class Vaccination

{

private:

int vacId;

char vacType[20];

public:

Vaccination();

Vaccination(int vId, char vType[]);

void displayVaccinationDetails();

~Vaccination();

};

class classType {

private:

int classTypeID;

string classTypeName;

public:

classType();

classType(int classID, string className);

void displayClassTypeDetails();

~classType();

};

class Flights

{

private:

int flightId;

char flightDestination[20];

public:

Flights();

Flights(int fID, char fDestination[]);

void displayFlightsDetails();

~Flights();

};

class Visa

{

private:

string visaID, visaType, issuedDate, expiredDate;

public:

Visa();

Visa(string vid, string vtype, string visdate, string vexdate);

string getVisaId();

string getVisaType();

string getIssuedDate();

string getExpiredDate();

void displayVisaDetails();

~Visa();

};

class Payment

{

private:

int paymentId;

char paymentMethod[20];

double amount;

public:

Payment();

Payment(int pid, char pmethod[], double amm);

void displayPaymentDetails();

~Payment();

};

Customer::Customer() {

customerName = "";

nic = "";

contactNo = "";

}

Customer::Customer(char cName[], char cID[], char nic[], double cNo) {

customerName = cName;

customerID = cID;

nic = nic;

contactNo = cNo;

}

char Customer::getCustomerName() {

return customerName;

}

char Customer::getNIC() {

return nic;

}

double Customer::getContactNo() {

return conctactNo;

}

void Customer::displayCustomerDetails() {

cout << "Customer ID :" << customerID << endl << "Customer Name :" << customerName << endl

<< "NIC : " << nic << endl << "Contact No : " << contactNo;

}

Customer::~Customer() {

}

Passport::Passport() {

strcpy(passportId, "");

strcpy(passportIssueDate, "");

strcpy(passportExpDate, "");

}

Passport::Passport(char pid[], char passIssDate[], char passExpDate[]) {

strcpy(passportId, pid);

strcpy(passportIssueDate, passIssDate);

strcpy(passportExpDate, passExpDate);

}

void Passport::displayPassportDetails() {

cout << "Passport ID : " << passportId << endl << "Passport Issue Date : " << passportIssueDate << endl

<< "Passport Expiry Date : " << passportExpDate;

}

Passport::~Passport() {

}

Tickets::Tickets() {

strcpy(ticketId, "");

strcpy(issuedDate, "");

strcpy(expiredDate, "");

strcpy(classType, "");

}

Tickets::Tickets(char tId[], char tIsDate[], char tExDate[], char cType[]) {

strcpy(ticketId, tId);

strcpy(issuedDate, tIsDate);

strcpy(expiredDate, tExDate);

strcpy(classType, cType);

}

void Tickets::displayTicketDetails() {

cout << "Ticket ID : " << ticketId << endl << "Issued Date : " << issuedDate << endl

<< "Expired Date : " << expiredDate << endl << "Class Type : " << classType;

}

Tickets::~Tickets() {

}

Vaccination::Vaccination() {

vacId = 0;

strcpy(vacType, "");

}

Vaccination::Vaccination(int vId, char vType[]) {

vacId = vId;

strcpy(vacType, vType);

}

void Vaccination::displayVaccinationDetails() {

cout << "Vaccination ID : " << vacId << endl << "Vaccination Type : " << vacType;

}

Vaccination::~Vaccination() {

}

classType::classType()

{

classTypeID = 001;

classTypeName = "sanath";

}

classType::classType(int classID, string className)

{

classTypeID = classID;

classTypeName = className;

}

void classType::displayClassTypeDetails()

{

cout << "class type id = " << classTypeID << endl;

cout << "class type name = " << classTypeName << endl;

}

classType::~classType() {

}

Flights::Flights() {

flightId = 0;

strcpy(flightDestination, "");

}

Flights::Flights(int fID, char fDestination[]) {

flightId = fID;

strcpy(flightDestination, fDestination);

}

void Flights::displayFlightsDetails() {

cout << "Vaccination ID : " << flightId << endl << "Vaccination Type : " << flightDestination;

}

Flights::~Flights() {

}

Visa::Visa() {

visaID = "";

visaType = "";

issuedDate = "";

expiredDate = "";

}

Visa::Visa(string vid, string vtype, string visdate, string vexdate) {

visaID = vid;

visaType = vtype;

issuedDate = visdate;

expiredDate = vexdate;

}

string Visa::getVisaId() {

return visaID;

}

string Visa::getVisaType() {

return visaType;

}

string Visa::getIssuedDate() {

return issuedDate;

}

string Visa::getExpiredDate() {

return visaID;

}

void Visa::displayVisaDetails() {

cout << "Visa ID : " << visaID << endl << "Visa Type : " << visaType << endl

<< "Visa Issued Date : " << issuedDate << endl << "Visa Expired Date : " << expiredDate;

}

Visa::~Visa() {

}

Payment::Payment() {

paymentId = 0;

strcpy(paymentMethod, "");

amount = 0.00;

}

Payment::Payment(int pid, char pmethod[], double amm) {

paymentId = pid;

strcpy(paymentMethod, pmethod);

amount = amm;

}

void Payment::displayPaymentDetails() {

cout << "Payment ID : " << paymentId << endl << "Payment Method : " << paymentMethod << endl

<< "Amount : " << amm;

}

Payment::~Payment() {

}

int main()

{

#include <iostream>

#include <cstring>

customer \*c1;

c1 = new customer("kamal", "123", "20011234455", "07763543211");

c1->displayCustomerDetails();

Passport \*p1;

p1 = new Passport("34536737", " 23 /l0 / 2018", " 23 /l0 / 2019");

p1->displayPassportDetails();

Tickets \*t1;

t1 = new Tickets("1345", "20 / 9 / 2022", "21 / 9 / 2022", "Economy");

t1->addReservation("25 / 9 / 2021");

t1->cancelReservation();

t1->displayTicketDetails();

Vaccination \*v1;

v1 = new Vaccination(01234, "CIP");

v1->displayVaccinationDetails();

classType \*c1;

c1 = new classType(3456, "Economy");

c1->displayClassTypeDetails();

Flights \*f1;

f1 = new Flights(7890, "colombo to india");

f1->displayFlightsDetails();

Visa \*vi1;

Visa = new vi1("1234", "Tourist", "23/ 10 / 2022", "23/ 11 / 2022");

v1->displayVisaDetails();

Payment \*p1;

p1 = new Payment(12334, "card", 20000.00);

p1->displayPaymentDetails();

delete c1;

delete p1;

delete t1;

delete v1;

delete c1;

delete f1;

delete vi1;

delete p1;

}