

Topic : Tourism and Travel Management System

Group no : MLB\_07.01\_10

Campus : Malabe / ~~Metro / Matara / Kandy / Kurunegala / Kandy / Jaffna~~

Submission Date: 19/05/2022

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** |
| IT21157332 | Sumanapala P.M.D.L.C |  |
| IT21156106 | D.M.S. Devinda |  |
| IT21106910 | Gimhani K.M.B.K |  |
| IT21157400 | Gangabadage U.D |  |
| IT21155970 | Madushan S.M.P.K.G.S |  |

Table of Contents

[1. Requirements 3](#_Toc103979790)

[2. Identified Classes 4](#_Toc103979791)

[3. CRC Cards 5](#_Toc103979792)

[4. Class Diagram (UML Notation) 8](#_Toc103979793)

[5. Cpp Files 9](#_Toc103979794)

# Requirements

* The system should function 24/7/365.
* There are two types of customers for the system, unregistered and registered customers.
* An unregistered customer can visit the website and view the travel packages that in the website.
* If they want to become a registered customer, they can become a registered customer by registering for the website.
* Registered customer can view the packages according to their perspectives.
* To register, they need to enter their personal details like name, contact details, email address and etc.
* Customer can also submit enquiries regarding the questions that they have about the packages and etc. Once they submit the enquiry they will receive a response through an email.
* After the registration, Customer can login to their account by using their username, password.
* Once they login to their account they can view packages and select any package according to them.
* After selecting their package, they will communicate through a travel agent about the travel via phone call or email.
* They can make the payment by using their card details.
* After the payment is confirmed by the bank or any other trusted resource they will receive an e-receipt about the details through an email.
* Administrator can add / update / delete details about the packages.
* Agency can create, customize travel plans and appoint travel agents to the customers.
* Once the customer went on the travel, they can post a review about their travel.

# Identified Classes

Registered customer class

Unregistered customer class

Administrator class

Agency class

Agent class

Package class

Transportation class

Booking class

Hotel class

Tour class

# CRC Cards

|  |  |
| --- | --- |
| **Class Name: Transportation** | |
| **Responsibilities** | **Collaborations** |
| Check transport methods | Customer |
| Check the details | Customer |
| Place a transport service | Agent |
| Calculate the total price | Travel agency |

|  |  |
| --- | --- |
| **Class Name: Booking** | |
| **Responsibilities** | **Collaborations** |
| Place a booking |  |
| Check the availability in hotel | Agent |
| Check the accommodation details | Customer |
| Calculate the total price | Hotel |

|  |  |
| --- | --- |
| **Class Name: Administrator** | |
| **Responsibility** | **Collaborators** |
| Add Package | package |
| Edit Package |  |
| Delete Package |  |
| Approve Booking | Administrator |
| Approve Agency | Booking |

|  |  |
| --- | --- |
| **Class Name: Agent** | |
| **Responsibility** | **Collaborators** |
| Add Agency |  |
| Edit Agency Details |  |
| Delete Agency | Administrator |

|  |  |
| --- | --- |
| **Class Name: Package** | |
| **Responsibilities** | **Collaborations** |
| Add Package | Administrator |
| Edit Package | Administrator |
| Delete Package | Administrator |
| Preview Selected Package | Booking |

|  |  |
| --- | --- |
| **Class Name: Agency** | |
| **Responsibilities** | **Collaborations** |
| Travel Payment | Registered Customer |
| Budget | Registered Customer |
| Travel Requested | Booking |
| Travel Approve | Agent |

|  |  |
| --- | --- |
| **Class Name: Unregistered Customer** | |
| **Responsibility** | **Collaborators** |
| Register to the system |  |
| Allow to view the Packages | Package |

|  |  |
| --- | --- |
| **Class Name: Registered Customer** | |
| **Responsibility** | **Collaborators** |
| Can view packages | Package |
| Add and update customer details |  |

|  |  |
| --- | --- |
| **Class Name: Hotel** | |
| **Responsibilities** | **Collaborations** |
| Hotel Register | Customer |
| Hotel booking | Booking |

|  |  |
| --- | --- |
| **Class Name: Tour** | |
| **Responsibilities** | **Collaborations** |
| Tour guide | Agent |
| Transporting and accompanying tourists | Transport |

# Class Diagram (UML Notation)

Diagram

Description automatically generated

# Cpp Files

#include <iostream>

#include<string.h> //string header file

#define SIZE

using namespace std;

//IT21157332

class RegisteredCustomer {

//attributes

protected:

string custUsername;

string custPassword;

private:

int packageID;

//methods

public:

RegisteredCustomer();

RegisteredCustomer(string rcustUsername, string rcustPassword, int rpackageID);

void addregistercustomer();

void login();

void displayDetails();

void logout();

void checkLogindetails();

void packageview();

};

//IT21157332

class UnregisteredCustomer {

//attributes

protected:

string custID;

string custName;

string custEmail;

string custAddress;

//methods

public:

UnregisteredCustomer();

UnregisteredCustomer(string ucustID, string ucustName, string ucustEmail, string ucustAddress);

void registercustomer();

void viewPackages();

void displayPackages();

};

//IT21157400

class Administrator { //class declaration

private: //set attributes

Administrator(int aadminID, string aadminName, string aadminEmail, string aadminUsername, string adminPassword);

int adminID{};

string adminName;

string adminEmail;

string adminUsername;

string adminPassword;

Package\*admin[SIZE];

public:

Administrator();//default constructor

Administrator(int aadminName,const string& aadminEmail,const string& aadminUsername,const string& adminPassword);//overload constructor

//methods

void addAdmin();

void editAdmin();

void deleteAdmin();

void updatePackage(Package\*admin);

void updatePackage();

~Administrator();

};

//IT21157400

class Agent { //class declaration

private: //set attributes

int agentID;

string agentName;

string agentEmail;

string agentUsername;

string agentPassword;

public: //methods

Agent();

Agent(int pagentId,string pagentName,string pagentContactNo,string pagentEmail,

string pagentUsername, string pagentPassword);

void addAgent();

void editAgent();

void deleteAgent();

~Agent(); //clear method

};

//IT21156106

class Agency {

//attributes

private:

int agencyId;

string agencyName;

string agencyType;

//methods

public:

Agency();

Agency(int aagencyId, string aagencyName, string aagencyType);

void TravelPayment();

void Budget();

void TravelRequsted();

void approveAgency();

};

//IT21156106

class Package {

//attributes

private:

int packageId;

string packageName;

string packageDescription;

string packageType;

float packagePrice;

//methods

public:

Package();

Package(int packId, string packName, string packType, string packDescription, float packPrice);

void addPackage();

void editPackage();

void deletePackage();

void displaySelectedPackage();

};

//IT21106910

class Transportation {

//attributes

private:

int transportation\_id;

int transportation\_tourid;

string transportation\_name;

string v\_type;

string v\_description;

//methods

public:

Transportation();

Transportation(int ttransportation\_id, int ttransportation\_tourid, string ttransportation\_name, string tv\_type, string tv\_description);

void addTransportation();

void editTransportation();

void deleteTransportation();

void updateTransportation();

void saveTransportation();

void searchTransportation();

void displayTransportationDetails();

};

//IT21106910

class Booking{

//attributes

private:

int booking\_id;

int booking\_hotelid;

string booking\_title;

string booking\_type;

string booking\_date;

string booking\_description;

//methods

public:

Booking();

Booking(int bbooking\_id, int bbooking\_hotelid, string bbooking\_title, string bbooking\_type, string bbooking\_date, string bbooking\_description);

void addBooking();

void editBooking();

void deleteBooking();

void updateBooking();

void saveBooking();

void searchBooking();

void displayBookingDetails();

float calcbooking\_total();

};

//IT21155970

class Hotel {

//attributes

private:

int HotelID;

string HotelName;

string HotelType;

string HotelAddress;

string HotelDescription;

//methods

public:

Hotel();

Hotel(int HID,string HName,string Htype,string HAddress, string HDescription);

void addHotel();

void editHotel();

void deleteHotel();

void updateHotel();

void saveHotel();

void searchHotel();

};

//IT21155970

class Tour {

//attributes

private:

int TourID;

string TourName;

string TourType;

string TourPlace;

string TourDescription;

//methods

public:

Tour();

Tour(int TID, string TName, string TType, string TPlace, string TDescription);

void addTour();

void editTour();

void deleteTour();

void updateTour();

void saveTour();

void searchTour();

};

//end of the creating classes

//IT21157332

//default constructor

RegisteredCustomer::RegisteredCustomer()

{

custUsername = " ";

custPassword = " ";

packageID = " ";

}

//overload constructor

RegisteredCustomer::RegisteredCustomer(string rcustUsername,string rcustPassword)

{

custUsername = rcustUsername;

custPassword = rcustPassword;

packageID = rpackageID;

}

void RegisteredCustomer::addRegisteredCustomer(){

}

void RegisteredCustomer::login(){

}

void RegisteredCustomer::displayDetails(){

}

void RegisteredCustomer::logout(){

}

void RegisteredCustomer::checkLogindetails(){

}

void RegisteredCustomer::packageview(){

}

//IT21157332

//default constructor

UnregisteredCustomer::UnregisteredCustomer()

{

custID = 0;

custName = " ";

custEmail = " ";

custAddress = " ";

}

//overload constructor

UnregisteredCustomer::UnregisteredCustomer(string ucustID, string ucustName, string ucustEmail, string ucustAddress)

{

custID = ucustID;

custName = ucustName;

custEmail = ucustEmail;

custAddress = ucustAddress;

}

void UnregisteredCustomer::registercustomer(){

}

void UnregisteredCustomer::viewPackages(){

}

void UnregisteredCustomer::displayPackages(){

}

//IT21157400

Administrator::Administrator() {

adminID=0;

adminName="";

adminEmail="";

adminUsername="";

}

Administrator::Administrator(int aadminID,string aadminName, string aadminEmail, string aadminUsername, string aadminPassword) {

adminID=aadminID;

adminName=std::move(aadminName);

adminEmail=std::move(aadminEmail);

adminUsername=std::move(aadminUsername);

adminPassword=std::move(aadminPassword);

}

//implement Method

void Administrator::addAdmin() {}

void Administrator::deleteAdmin() {}

void Administrator::editAdmin() {}

void Administrator::updatePackage() {}

Administrator::Administrator(int aadminName, const string& aadminEmail, const string& aadminUsername, const string& adminPassword) {

}

void Administrator::updatePackage(Package \*admin) {

}

Administrator::~Administrator(){}

//IT21157400

Agent::Agent()

{

agentID= 0;

agentName = "";

agentContactNo = "";

agentEmail = "";

agentUsername = "";

agentPassword = "";

}

Aagent::Aagent(int aagentID,string aagentName, string aagentContactNo, string aagentEmali ,string aagentUsername, string aagentPassword)

{

agentID = aagentID;

agentName = aagentName;

agentContactNo = aagentContactNo;

agentEmail = aagentEmali;

agentUsername = aagentUsername;

agentPassword = aagentPassword;

}

void Agent::addAgent() {}

void Agent::deleteAgent() {}

void Agent::editAgent() {}

Agent::~Agent() {}

//IT21156106

Agency::Agency()

{

agencyId= 0;

agencyName = "";

agencyType = "";

}

//overload constructor

Agency::Agency(int aagencyId, string aagencyName, string aagencyType)

{

agencyId = aagencyId;

agencyName = aagencyName;

agencyType = aagencyType;

}

void Agency::TravelPayment() {

}

void Agency::Budget() {

}

void Agency::TravelRequsted() {

}

void Agency::approveAgency() {

}

//IT21155106

Package::Package()

{

packageId= 0;

packageName = "";

packageType = "";

packageDescription = "";

packagePrice = 0;

}

Package::Package(int packId, string packName, string packType, string packDescription, float packPrice)

{

packageId = packId;

packageName = packName;

packageType = packType;

packageDescription = packDescription;

packagePrice = packPrice;

}

void Package::addPackage() {

cout<<"Package Added"<<endl;

}

void Package::editPackage() {

cout<<"Edited"<<endl;

}

void Package::deletePackage() {

cout<<"Deleted"<<endl;

}

void Package::displaySelectedPackage() {

cout<<"Display Package"<<endl;

}

//IT21106910

//default constructor

Transportation::Transportation()

{

transportation\_id = 0;

transportation\_tourid = 0;

transportation\_name = "";

v\_type = "";

v\_description = "";

}

//overload constructor

Transportation::Transportation(int ttransportation\_id, int ttransportation\_tourid, string ttransportation\_name, string tv\_type, string tv\_description)

{

transportation\_id = ttransportation\_id;

transportation\_tourid = ttransportation\_tourid;

transportation\_name = ttransportation\_name;

v\_type = tv\_type;

v\_description = tv\_description;

}

void Transportation::addTransportation(){

}

void Transportation::editTransportation(){

}

void Transportation::deleteTransportation(){

}

void Transportation::updateTransportation(){

}

void Transportation::saveTransportation(){

}

void Transportation::searchTransportation(){

}

void Transportation::displayTransportationDetails(){

}

//IT21106910

Booking::Booking()

{

booking\_id = 0;

booking\_hotelid = 0;

booking\_title = "";

booking\_type = "";

booking\_date = "";

booking\_description = "";

}

Booking::Booking(int bbooking\_id, int bbooking\_hotelid, string bbooking\_title, string bbooking\_type, string bbooking\_date, string bbooking\_description)

{

booking\_id = bbooking\_id;

booking\_hotelid = bbooking\_hotelid;

booking\_title = bbooking\_title;

booking\_type = bbooking\_type;

booking\_date = bbooking\_date;

booking\_description = bbooking\_description;

}

void Booking::addBooking(){

}

void Booking::editBooking(){

}

void Booking::deleteBooking(){

}

void Booking::updateBooking(){

}

void Booking::saveBooking(){

}

void Booking::searchBooking(){

}

void Booking::displayBookingDetails(){

}

float Booking::calcbooking\_total(){

}

//IT21155970

Hotel::Hotel()

{

HotelID= 0;

HotelName = "";

HotelType = "";

HotelAddress = "";

HotelDescription = "";

}

Hotel::Hotel(int HID,string HName,string Htype,string HAddress, string HDescription)

{

HotelID =HID ;

HotelName=HName;

HotelType=Htype;

HotelAddress=HAddress;

HotelDescription=HDescription;

}

void Hotel::addHotel()

{

}

void Hotel::editHotel()

{

}

void Hotel::deleteHotel()

{

}

void Hotel::updateHotel()

{

}

void Hotel::saveHotel()

{

}

void Hotel::searchHotel()

{

}

//IT21155970

Tour::Tour()

{

TourID= 0;

TourName = "";

TourType = "";

TourPlace = "";

TourDescription = "";

}

Tour::Tour(int TID, string TName, string TType, string TPlace, string TDescription)

{

TourID =TID ;

TourName=TName;

TourType=TType;

TourPlace=TPlace;

TourDescription=TDescription;

}

void Tour::addTour()

{

}

void Tour::editTour()

{

}

void Tour::deleteTour()

{

}

void Tour::updateTour()

{

}

void Tour::saveTour()

{

}

void Tour::searchTour()

{

}

int main() {

//IT21157332

RegisteredCustomer r1, r2, r3;

//calling addRegisteredCustomer() method

r1.addRegisteredCustomer();

r2.addRegisteredCustomer();

r3.addRegisteredCustomer();

//calling login() method

r1.login();

r2.login();

r3.login();

//calling displayDetails() method

r1.displayDetails();

r2.displayDetails();

r3.displayDetails();

//calling logout() method

r1.logout();

r2.logout();

r3.logout();

//calling checkLogindetails() method

r1.checkLogindetails();

r2.checkLogindetails();

r3.checkLogindetails();

//calling packageview() method

r1.packageview()();

r2.packageview()();

r3.packageview()();

//IT21157332

RegisteredCustomer r1, r2, r3;

//calling registercustomer() method

r1.registercustomer();

r2.registercustomer();

r3.registercustomer();

//calling viewPackages() method

r1.viewPackages();

r2.viewPackages();

r3.viewPackages();

//calling displayPackages() method

r1.displayPackages();

r2.displayPackages();

r3.displayPackages();

//IT21157400

Administrator admin1,admin2;

//calling addAdmin() method

admin1.addAdmin();

admin2.addAdmin();

//calling editAdmin() method

admin1.editAdmin();

admin2.editAdmin();

//calling deleteAdmin() method

admin1.deleteAdmin();

admin2.deleteAdmin();

//calling updatePackage() method

admin1.updatePackage();

admin2.updatePackage();

//IT21157400

Agent agnt1,agnt2,agnt3;

//calling addAgency() method

agnt1.addAgent();

agnt2.addAgent();

agnt3.addAgent();

//calling addAgency() method

agnt1.editAgent();

agnt2.editAgent();

agnt3.editAgent();

//calling addAgency() method

agnt1.deleteAgent();

agnty2.deleteAgent();

agnty3.deleteAgent();

//IT21156106

Agency agency1,agency2,agency3;

//calling TravelPayment() method

agency1.TravelPayment();

agency2.TravelPayment();

agency3.TravelPayment();

//calling Budget() method

agency1.Budget();

agency2.Budget();

agency3.Budget();

//calling TravelRequsted() method

agency1.TravelRequsted();

agency2.TravelRequsted();

agency3.TravelRequsted();

//calling approveAgency() method

agency1.approveAgency();

agency2.approveAgency();

agency3.approveAgency();

//IT21156106

Package pack1,pack2,pack3;

//calling addPackage method

pack1.addPackage();

pack2.addPackage();

pack3.addPackage();

//calling editPackage() method

pack1.editPackage();

pack2.editPackage();

pack3.editPackage();

//calling deletePackage() method

pack1.deletePackage();

pack2.deletePackage();

pack3.deletePackage();

//calling displaySelectedPackage() method

pack1.displaySelectedPackage();

pack2.displaySelectedPackage();

pack3.displaySelectedPackage();

//IT21106910

Transportation t1, t2, t3;

//calling addTransportation() method

t1.addTransportation();

t2.addTransportation();

t3.addTransportation();

//calling editTransportation() method

t1.editTransportation();

t2.editTransportation();

t3.editTransportation();

//calling deleteTransportation() method

t1.deleteTransportation();

t2.deleteTransportation();

t3.deleteTransportation();

//calling updateTransportation() method

t1.updateTransportation();

t2.updateTransportation();

t3.updateTransportation();

//calling saveTransportation() method

t1.saveTransportation();

t1.saveTransportation();

t1.saveTransportation();

//calling searchTransportation() method

t1.searchTransportation();

t2.searchTransportation();

t3.searchTransportation();

//calling displayTransportationDetails() method

t1.displayTransportationDetails();

t2.displayTransportationDetails();

t3.displayTransportationDetails();

//IT21106910

Booking b1, b2, b3;

//calling addBooking() method

b1.addBooking();

b2.addBooking();

b3.addBooking();

//calling editBooking() method

b1.editBooking();

b2.editBooking();

b3.editBooking();

//calling deleteBooking() method

b1.deleteBooking();

b2.deleteBooking();

b3.deleteBooking();

//calling updateBooking() method

b1.updateBooking();

b2.updateBooking();

b3.updateBooking();

//calling saveBoking()method

b1.saveBooking();

b2.saveBooking();

b3.saveBooking();

//calling searchBookng() method

b1.searchBooking();

b2.searchBooking();

b3.searchBooking();

//calling displayBookingDetails() method

b1.displayBookingDetails();

b2.displayBookingDetails();

b3.dispalyBookingDetails();

for(int i = 1; i <= 3; i++)

{

cout << "Total Amount for booking" << i << "=" << calcbooking\_total() << endl;

}

//IT21155970

Hotel H1, H2, H3;

//calling addHotel method

H1.addHotel();

H2.addHotel();

H3.addHotel();

//calling editHotel method

H1.editHotel();

H2.editHotel();

H3.editHotel();

//calling deleteHotel method

H1.deleteHotel();

H2.deleteHotel();

H3.deleteHotel();

//calling updateHotel method

H1.updateHotel();

H2.updateHotel();

H3.updateHotel();

//calling saveHotelmethod

H1.saveHotel();

H2.saveHotel();

H3.saveHotel();

//calling searchHotel method

H1.searchHotel();

H2.searchHotel();

H3.searchHotel();

//IT21155970

Tour T1, T2, T3;

//calling addTour method

T1.addTour();

T2.addTour();

T3.addTour();

//calling editTour method

T1.editTour();

T2.editTour();

T3.editTour();

//calling deleteTour method

T1.deleteTour();

T2.deleteTour();

T3.deleteTour();

//calling updateTour method

T1.updateTour();

T2.updateTour();

T3.updateTour();

//calling saveTour method

T1.saveTour();

T2.saveTour();

T3.saveTour();

//calling searchTour method

T1.searchTour();

T2.searchTour();

T3.searchTour();

Transportation t1, t2, t3;

//calling addTransportation() method

t1.addTransportation();

t2.addTransportation();

t3.addTransportation();

//calling editTransportation() method

t1.editTransportation();

t2.editTransportation();

t3.editTransportation();

//calling deleteTransportation() method

t1.deleteTransportation();

t2.deleteTransportation();

t3.deleteTransportation();

//calling updateTransportation() method

t1.updateTransportation();

t2.updateTransportation();

t3.updateTransportation();

//calling saveTransportation() method

t1.saveTransportation();

t1.saveTransportation();

t1.saveTransportation();

//calling searchTransportation() method

t1.searchTransportation();

t2.searchTransportation();

t3.searchTransportation();

//calling displayTransportationDetails() method

t1.displayTransportationDetails();

t2.displayTransportationDetails();

t3.displayTransportationDetails();

return 0;

}