



Topic : Hotel Reservation System for Tourists

Group no : MLB_04.01_01

Campus : Malabe

Submission Date : 18/05/2022

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.

	Student ID	Student Name	Email	Contact Number
1	IT21251900	Rajapaksha R.M.S.D	It21251900@my.sliit.lk	0765491514
2	IT21251078	Weerasinghe W.M.V.S	It21251078@my.sliit.lk	0711810057
3	IT21251214	Manathunga A.U.U	It212251214@my.sliit.lk	0764509902
4	IT21249884	Ranathunga R.V.R.N.U	It21249884@my.sliit.lk	0711212914
5	IT21250224	Neluvinda S.P.S.M	It21250224@my.sliit.lk	0777553619

Description

Tugo is an Online Hotel Reservation System for the tourists who visit Sri Lanka. It allows the tourists to browse and reserve hotels. Also, it provides online hotel booking experience to all its customers with a user-friendly interface and a fast access. Tourists can just visit the website without any need of login and browse the hotels according to their preferences.

Tugo provides all the information regarding the best and popular hotels in Sri Lanka. Any customer can search hotels at their favorite locations at a very reasonable price.

Our goal is to build a website to provide great and easy hotel booking experiences to travelers, lovers, families, and all the other users while providing good services and customer support. The tourists need to register and login to the platform only when they are going to reserve a hotel. To reserve a hotel, they must provide following information: "Name, Contact Number, Email Address, Check in date, Check out date, No. of guests in relevant hotel reservation page. After providing those details user can confirm booking and make necessary payments.

It is vital to state that we are reliable for all the information regarding the online reservation purposes and every customer can trust the platform.

Not only for the customers, the hotel managers also can login to the website as managers and add their hotels to the system. When adding new hotels, they also have to provide "hotel name, location, contact details, amenities, prices and hotel photos".

Another special feature of this website is, every user, every hotel and all the reservations have separate unique ID's. User registrations and hotel manager registrations are both free of charge. However, when reservation happens site collect 2% of the transaction as a commission. That is the sites revenue model.

Requirements

- Register the system as a new user
- Register as a new hotel
- Cancel registration
- Login to the system as a registered user
- Login to the system as a hotel
- Login to the system as the system admin
- View the recommended hotels
- Search a hotel as a registered user
- Search a hotel as a guest.
- Reserve a hotel as a registered user
- Enter reservation details
- Enter payment details
- Confirm reservations
- Check/view the reservations as a registered user
- Cancel reservations as a registered user
- Update user profile as a registered user
- Manage reviews as a registered user
- View reservation as a hotel
- View reviews as a hotel
- Update hotel details as a hotel
- Manage user as the system admin
- Manage hotel as the system admin
- Control site information as the system admin
- Check password validity
- Validate email addresses and contact numbers
- Validate user
- Logout from the system

Classes

- Customer
- Hotel
- Admin
- Review
- Site settings
- Payment
- reservation

CRC Cards

customer	
Responsibility	Collaborators
Register	
Cancel registration	
Login to the system	
View hotel	hotel
Search a hotel	hotel
Reserve a hotel	Hotel, reservation, payment
View reservation	reservation
Cancel reservation	Reservation,hotel,payment
Update customer profile details	
Manage reviews	Review,hotel

admin	
Responsibility	Collaborators
Login to the system	
Manage customer	customer
Manage hotel	hotel
Control site information	Site settings

review	
Responsibility	Collaborators
Add review	
Update review	
Delete review	

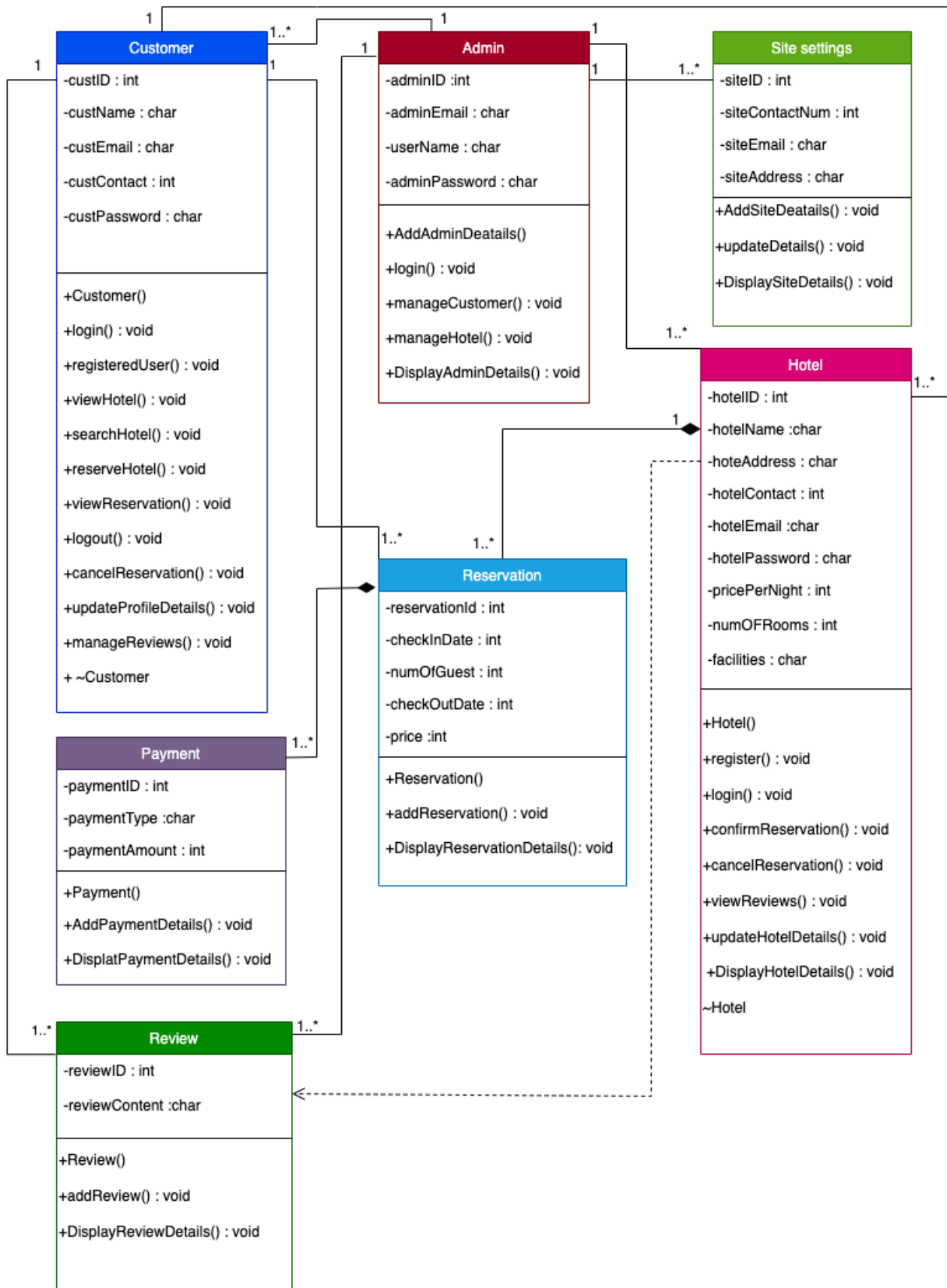
reservation	
Responsibility	Collaborators
Enter reservation details	Hotel, customer

hotel	
Responsibility	Collaborators
Register	
Cancel registration	
Login to the system	
Confirm reservation	reservation
Cancel reservation	Reservation,hotel,payment
View reviews	review
Update hotel details	

Site settings	
Responsibility	Collaborators
Add details	
Update detail	

payment	
Responsibility	Collaborators
Enter payment details	reservation

Class diagram



C++ Codes

1. IT21251900

Hotel.h

```
#include "Admin.h"
#include "Customer.h"
#include "Reservation.h"
#include "Review.h"
#define SIZE 2

class Hotel {
private:
    int hotelID;
    char hotelName[50];
    char hotelAddress[256];
    int hotelContact;
    char hotelEmail[256];
    int pricePerNight;
    int numOfRooms;
    char facilities[1000];
    Customer *cust;
    Reservation *res[SIZE];
    Admin *ad;

public:
    Hotel();

    Hotel(
        int PhotelID,
        char PhotelName[],
        char PhotelAddress[],
        int PhotelContact,
        char PhotelEmail[],
        int PpricePerNight,
        int PnumOfRooms,
        char Pfacilities[],
        Customer *Pcust,
        Admin *Pad);

    Hotel(
        int ReservationID1,
        int CheckInDate1,
        int CheckOutDate1,
        int NoOfGuests1,
        int NoOfNights1,
        int ReservationID2,
        int CheckInDate2,
        int CheckOutDate2,
        int NoOfGuests2,
        int NoOfNights2);

    void addReview(Review * rev);
    void login();
    void confirmReservation();
    void addReview(Review*rev);
    void cancelReservation();
    void updateHotelDetails();
    void DisplayHotelDetails();

    ~Hotel();
};
```

1. IT21251900

Hotel.cpp

```
#include <iostream>
#include <cstring>
#include "Hotel.h"

Hotel::Hotel(){}

Hotel::Hotel(int PhotoID, char PhotoName[], char PhotoAddress[], int PhotoContact, char PhotoEmail[],
             int PpricePerNight,int PnumOfRooms, char Pfacilities[],Customer *Pcust, Admin *Pad){

    hotelID=PhotoID;
    strcpy(hotelName,PhotoName);
    strcpy(hotelAddress,PhotoAddress);
    hotelContact=PhotoContact;
    strcpy(hotelEmail,PhotoEmail);
    pricePerNight=PpricePerNight;
    numOfRooms=PnumOfRooms;
    strcpy(facilities,Pfacilities);
    cust=Pcust;
    ad=Pad;

}

Hotel::Hotel(int ReservationID1, int CheckInDate1, int CheckOutDate1, int NoOfGuests1, int NoOfNights1,
             int ReservationID2, int CheckInDate2, int CheckOutDate2, int NoOfGuests2, int NoOfNights2){

    res[0]=new Reservation(int ReservationID1, int CheckInDate1, int CheckOutDate1, int NoOfGuests1, int NoOfNights1);
    res[1]=new Reservation(int ReservationID2, int CheckInDate2, int CheckOutDate2, int NoOfGuests2, int NoOfNights2);
}

void Hotel:: addReview(Review * rev){

}

void Hotel::login(){

}

void Hotel:: confirmReservation(){

}

void Hotel::addReview(Review*rev){

}

void Hotel::cancelReservation(){

}

void Hotel::updateHotelDetails(){

}

void Hotel::DisplayHotelDetails(){

}

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

2. IT21251078

Payment.h

```
class Payment{
private:
    int paymentID;
    char paymentType[10];
    int paymentAmount;
public:
    Payment();
    void addPaymentDetails(int p_ID, const char p_type[], int p_amount);
    void displayPaymentDetails();
};
```

Payment.cpp

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

Payment::Payment(){

}

void Payment::addPaymentDetails(int p_ID, const char p_type[], int p_amount){
    paymentID = p_ID;
    strcpy(paymentType,p_type);
    paymentAmount = p_amount;
}

void Payment::displayPaymentDetails(){
    cout << "Payment ID : " << paymentID << endl
    << "Payment Type : " << paymentType<< endl
    << "Payment Amount: " << paymentAmount << endl;
}
```

Review.h

```
class Review{
private:
    int reviewID;
    char reviewContent[50];
public:
    void AddReviewDetails( int r_ID, char r_Content[]);
    void DisplayReviewDetails();
};
```

Review.cpp

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

void Review::AddReviewDetails(int r_ID, char r_Content[])
{
    reviewID = r_ID;
    strcpy(reviewContent, r_Content);
}

void Review::DisplayReviewDetails()
{
    cout << "Review ID: " << reviewID << endl
    << "Review Content : " << reviewContent << endl;
}
```


3. IT21251214

Admin.h

```
class SiteSettings;  
class Hotel;  
class Review;
```

```
class Customer;  
class Hotel;  
class Review;  
class SiteSettings;
```

```
#define SIZE 10;
```

```
        class Admin{  
private:  
    int adminID;  
    char adminEmail[50];  
    char userName[50];  
    char adminPassword[50];  
    SiteSettings *settings[SIZE];  
    Hotel *hotels[SIZE];  
    Review *reviews[SIZE];  
    Customer *customers[SIZE];  
  
public:  
    Admin();  
    Admin(int pID, const char pEmail[], const char pUsername[], const char pPassword[]);  
    void DisplayAdminDetails();  
    void login();  
    void manageCustomers();  
    void manageHotels();  
    ~Admin();  
};
```

3. IT21251214

Admin.cpp

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

Admin::Admin(){

}

Admin::Admin(int pID, const char pEmail[], const char pUsername[], const char pPassword[])
{
    adminID = pID;
    strcpy(adminEmail, pEmail);
    strcpy(userName, pUsername);
    strcpy(adminPassword, pPassword);
}

void Admin::DisplayAdminDetails()
{
    cout << "Admin ID : " << adminID << endl
         << "Admin email : " << adminEmail << endl
         << "Username : " << userName << endl
         << "Password : " << adminPassword << endl;
}

void Admin::login(){

}

void Admin::manageCustomers(){

}

void Admin::manageHotels(){

}

Admin::~Admin(){

}
```

3. IT21251214

SiteSetting.h

```
class Admin;

class SiteSettings
{
private:
    int siteID;
    int siteContactNum;
    char siteEmail[50];
    char siteAddress[50];
    Admin *a;

public:
    SiteSettings();
    SiteSettings(int pld, int pNum, const char pEmail[], const char pAddress[], Admin *pa);
    void DisplaySiteDetails();
    void updateDetails();
    ~SiteSettings();
};
```

SiteSetting.cpp

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

SiteSettings::SiteSettings(){}

SiteSettings::SiteSettings(int pld, int pNum, const char pEmail[], const char pAddress[], Admin *pa)
{
    siteID = pld;
    siteContactNum = pNum;
    strcpy(siteEmail, pEmail);
    strcpy(siteAddress, pAddress);
}

void SiteSettings::DisplaySiteDetails()
{
    cout << "Site ID: " << siteID << endl
         << "Site Contact Number : " << siteContactNum << endl
         << "Site email : " << siteEmail << endl
         << "Site address : " << siteAddress << endl;
}

void SiteSettings::updateDetails(){

}

SiteSettings::~~SiteSettings(){

}
```

4. IT21249884

Reservation.h

```
#include "Payment.h"
#define SIZE 2;

class Reservation{
protected:
    int reservationId;

private:
    int checkInDate;
    int checkOutDate;
    int noOfGuests;
    int price;
    int noOfNights;

    //Composition relationship with "Payment class"
    Payment *pay[SIZE];

public:
    Reservation(); //Default Constructor
    Reservation:: Reservation(int ReservationID, int CheckInDate, int CheckOutDate, int NoOfGuests, int NoOfNights, );
//Overloading Constructor
    Reservation::Reservation(int payment1, int payment2);
    void DisplayReservationDetails();
    ~Reservation();
};
```

4. IT21249884

Reservation.cpp

```
#include<iostream>
#include "Reservation.h"
#include "Payment.h"
#define SIZE 2;

using namespace std;

//Implementation of the default constructor - Assignning default values
Reservation:: Reservation(){
    reservationId = 0;
    checkInDate = 0;
    checkOutDate = 0;
    noOfGuests = 0;
    noOfNights = 0;
}

//Implementation of the overloading constructor
Reservation:: Reservation(int ReservationID, int CheckInDate, int CheckOutDate, int NoOfGuests, int NoOfNights){
    reservationId = ReservationID;
    checkInDate = CheckInDate;
    checkOutDate = CheckOutDate;
    noOfGuests = NoOfGuests;
    noOfNights = NoOfNights;
}

Reservation:: Reservation(int payment1, int payment2){
    for(int i=0;i<SIZE;i++){
        pay[i]=new payment(payment1);
        pay[1]=new payment(payment2);
    }
}

//Displaying the reservation details
void Reservation:: DisplayReservationDetails(){
    cout << "Reservation Id   :" << reservationId << endl;
    cout << "CheckInDate       :" << reservationId << endl;
    cout << "CheckOutDate        :" << reservationId << endl;
    cout << "Number Of guests   :" << reservationId << endl;
    cout << "Number Of nights  :" << reservationId << endl;

    for ( int i = 0 ; i < SIZE ; i++ )
        payment[i] -> DisplayPayment();
}

//Implementation of the destructor
Reservation:: ~Reservation(){
    cout << "Deleting the reservation..." << endl;

    for ( int i=0 ; i < SIZE ; i++)
        delete payment[i];
}
```

Customer.h

```
#include "Hotel.h"
#include "Reservation.h"
#include "Admin.h"
#include "Review.h"
#define SIZE 10;

class customer{
private:
    int custID;
    char custName[10];
    char custEmail[10];
    int custContact;
    char custPassword;
    int noOfReviews;
    int noOfReservations;
    Hotel* hotel[SIZE];
    Review* rev[SIZE];
    Admin* ad;
    Reservation* res[SIZE];

public:
    customer();
    customer(int cID, char cName[], char ctEmail[], int cContact, char cPassword[], Admin* Pad, Hotel* Phot[SIZE],
            int PnoOfReviews, int PnoOfReservations);
    void addReview(Review* Prev);
    void addReservation(Reservation* Pres);
    void login();
    void viewHotel();
    void searchHotel();
    void logout();
    void viewReservation();
    void cancelReservation();
    void updateProfileDetails();
    ~customer();
}
```

Customer.cpp

```
#include"Customer.h"
#include<cstring>
//customer s
customer::customer(){}
customer::customer(int cID,char cName[],char ctEmail[],int cContact,char cPassword[],Admin*Pad,
                  Hotel*Phot[SIZE],int PnoOfReviews,int PnoOfReservations)
{
    custID=cID;
    strcpy(custName,cName);
    strcpy(custEmail,ctEmail);
    custContact=cContact;
    noOfReviews=PnoOfReviews;
    noOfReservations=PnoOfReservations;
    ad=Pad;
    for(int i=0;i<SIZE;i++){
        hot[i]=Phot[i];
    }
}
void customer:: addReview(Review*Prev){
    if(noOfReviews<SIZE){
        rev[noOfReviews]=Prev;
        noOfReviews++;
    }
}
void customer::addReservation(Reservation*Pres){
    if(noOfReservations<SIZE){
        rev[noOfReservations]=Pres;
        noOfReviews++;
    }
}
void customer::login(){

}
void customer::viewHotel(){

}
void customer::searchHotel(){

}
void customer::logout(){

}
void customer::viewReservation(){

}
void customer::cancelReservation(){

}
void customer::updateProfileDetails(){

}
customer::~customer(){

}
```

main.cpp

```
#include <cstring>
#include <iostream>

#include "Admin.h"
#include "Customer.h"
#include "Hotel.h"
#include "Reservation.h"
#include "siteSettings.h"

using namespace std;

int main() {

    Hotel *hotel;
    Admin *a1;
    SiteSettings *s1;
    Reservation *R1;
    customer *c;

    hotel = new Hotel();
    a1 = new Admin(1, "admin@gmail.com", "admin22", "password");
    s1 = new SiteSettings(1, 0777777777, "tugo@gmail.com", "no:21,address", a1);
    R1 = new Reservation();
    c = new customer();

    hotel->login();
    hotel->confirmReservation();
    hotel->addReview(Review * rev);
    hotel->cancelReservation();
    hotel->updateHotelDetails();
    hotel->DisplayHotelDetails();

    a1->DisplayAdminDetails();
    s1->DisplaySiteDetails();

    R1->DisplayReservationDetails();

    c->login();
    c->logout();
    c->cancelReservation();
    c->searchHotel();
    c->addReservation();
    c->addReview();
    c->updateProfileDetails();
    c->viewHotel();
    c->viewReservation();

    delete hotel;
    delete a1;
    delete s1;
    delete c;

    return 0;
}
```


Individual Contribution

	Student ID	Student Name	Individual Contribution
1	IT21251900	Rajapaksha R.M.S.D	<ul style="list-style-type: none">• Hotel class
2	IT21251078	Weerasinghe W.M.V.S	<ul style="list-style-type: none">• Payment class• Review class
3	IT21251214	Manathunga A.U.U	<ul style="list-style-type: none">• Admin class• Site settings class
4	IT21249884	Ranathunga R.V.R.N.U	<ul style="list-style-type: none">• Reservation class
5	IT21250224	Neluvinda S.P.S.M	<ul style="list-style-type: none">• Customer class