



Topic : Hotel Reservation System for Special Events

Group no : MLB_WE_01.01.03

Campus : Malabe

Submission Date: 19th May 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
IT21328770	Kulathilake A.M.M	0760341842
IT21338748	De Silva B.A. P	0779403056
IT21279584	Sudusinghe H. J	0722610353
IT21262340	Rathnayake L.D.C	0768335083
IT21278280	Fernando W. T. R. P	0769782381

Contribution:

Registration No	Name	Contribution
IT21328770	Kulathilake A.M.M	<ul style="list-style-type: none">• Customer and feedback• Requirements
IT21338748	De Silva B.A. P	<ul style="list-style-type: none">• Payment and Discount• Main.cpp
IT21279584	Sudusinghe H. J	<ul style="list-style-type: none">• Reservation and Hotel Hall• Identify Nouns and classes
IT21262340	Rathnayake L.D.C	<ul style="list-style-type: none">• Sky Hall, Ball Hall and Gold Hall• Description
IT21278280	Fernando W. T. R. P	<ul style="list-style-type: none">• Reservation, Receptionist and Receipt

Hotel Reservation System for Special Events – Peppers Grand Hotel

Hotel reservations are made through an online system that allows you to make any type of reservation without having to visit the hotel. Customers can make safe and secure reservations online with this method. When a customer logs in, the system displays information and all the data regarding the various types of special event reservations.

If a customer wishes to make a reservation, they must enter personal details and reservation details and confirm it. After the confirmation hotel issues the customer with customer ID.

The hotel will provide the customer verification details by email or text message once the payment is completed. The customer ID, name, receipt number, amount, reserve date, and staff ID are all listed on this receipt. Customer can pay using credit cards, debit cards or PayPal methods.

The customer also can cancel his or her reservation.

The receptionist and administrator can also use his or her username and password to create and edit reservations, as well as enter new booking details.

At the start of the booking process, the receptionist collects data from the customer and delivers it to the manager. The database administrator receives the relevant information from the manager. And the customer details are saved in the system. The customer ID is used to store each customer's information.

Requirements:

As a customer, first I must log in to make a reservation for halls/rooms.

Customers can check whether rooms and halls are available on the desired date and time.

Customers can reserve by providing the information below. Name, NIC, Address, E-mail and phone number.

Customer can correct any errors in the registration details. Also, customer can change and cancel the reservation.

When a customer modifies their booking, the system updates the accommodation details and customer's record.

The payment will be displayed, and customer will be able to pay by card.

The payment details will be verified by the system.

Then the booking confirmation is sent by the system.

To prevent unauthorized users from changing or editing booking details, the system encrypts data.

With the customer's request, the receptionist enters the new details (updating check in, check out, room/hall choices, and cancellation) into the system, and the receptionist can edit the customer's booking information. (Name, employee number, and email address)

Update the reservation database with the new item information and records.

Customer can provide reviews on the service to the system, and the system will respond. The database administrator will correct any errors that occur.

Noun:

- Customer
- Date
- Time
- Hall
- Email
- Name
- NIC
- Payment
- Amount
- Message
- Record
- Reservation
- Description
- Database Administrator
- Receptionist
- Item Details
- Customer Details
- Payment Details
- Ballhall
- Skyhall
- Goldhall
- PayPal
- Card
- Staff Email
- Feedback
- Feedback ID
- Staff ID
- Discount

Classes:

- Customer
- Halls
- BallHall
- SkyHall
- GoldHall
- Receipt
- Payment
- Receptionist
- Feedback
- Reservation
- Discount

CRC Card:

Class Name: Customer	
Relationship:	Collaboration:
Login to the system and check availability of the hall	Halls
Enter details and confirm	
Provide feedback about the service	

Class Name: Halls	
Relationship:	Collaboration:
Provide package details	
Check availability of halls	Customer

Class Name: BallHall	
Relationship:	Collaboration:
Providing package details	

Class Name: SkyHall	
Relationship:	Collaboration:
Providing package details	

Class Name: GoldHall	
Relationship:	Collaboration:
Providing package details	

Class Name: Receptionist	
Relationship:	Collaboration:
Login to the system	
Edit bookings according to request	Customer

Class Name: Receipt	
Relationship:	Collaboration:
Generate the receipt	Customer, Receptionist, Payment, Halls

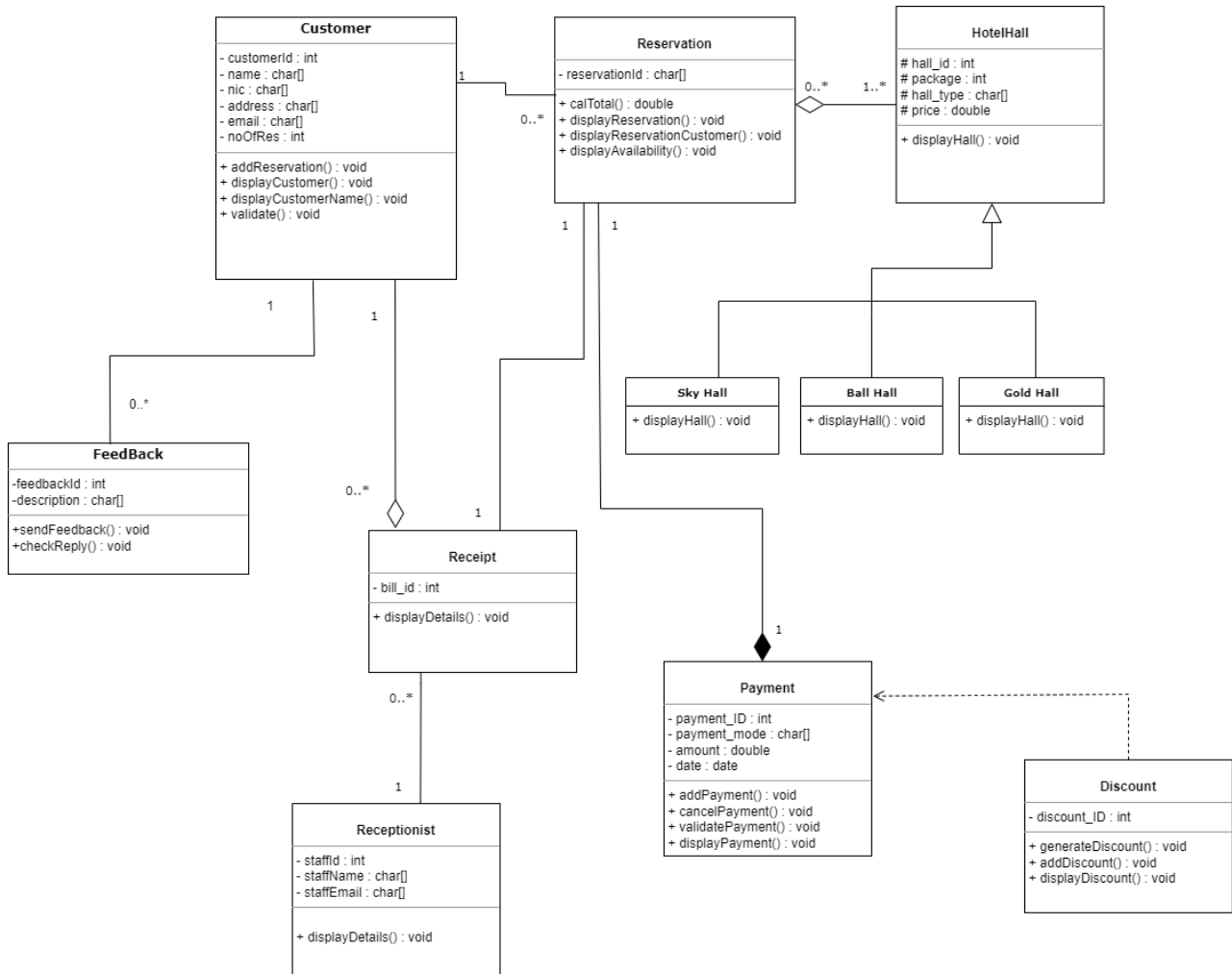
Class Name: Payment	
Relationship:	Collaboration:
Calculate the total payment	Halls

Class Name: Feedback	
Relationship:	Collaboration:
Send the feedback	Customer
Check the reply	

Class Name: Discount	
Relationship:	Collaboration:
Generate Discount	Payment
Calculate discount	

Class Name: Reservation	
Relationship:	Collaboration:
Making reservations	Customer, Halls

UML class diagram:



Main.cpp

```
#include <iostream>
#include "Customer.h"
#include "Discount.h"
#include "Payment.h"
#include "HotelHall.h"
#include "GoldHall.h"
#include "SkyHall.h"
#include "BallHall.h"
#include "Feedback.h"
#include "Receipt.h"
#include "Reservation.h"
#include "Receptionist.h"

using namespace std;

int main()
{
    //customer constructor
    Customer* C1 = new Customer(001, "Rusith",
    "99457890345v", "rusith123@gmail.com", "Wennappuwa");
    Customer* C2 = new Customer(002, "Hansaka",
    "8763456789v", "hansaka@gmail.com", "Kottawa");

    //discount constructor
    Discount* D1 = new Discount(001);
    Discount* D2 = new Discount(002);

    //Feedback constructor
    Feedback* F1 = new Feedback(001, "Good");
    Feedback* F2 = new Feedback(002, "GoodService");

    //Payment constructor
    Payment* P1 = new Payment(001, "Credit Card",
    100000, "2022-06-02");
    Payment* P2 = new Payment(002, "Debit Card",
    175000, "2022-07-22");
```

```

//Receipt constructor
Receipt* R1 = new Receipt();

//Receptionist constructor
Receptionist* RP1 = new Receptionist(006, "Amanda",
"amanda123@gmail.com");
Receptionist* RP2 = new Receptionist(007, "Menula",
"menula123@gmail.com");

//Reservation constructor
Reservation* RV1 = new Reservation();

//HotelHall constructor
HotelHall* H1 = new HotelHall(004, 2, "SkyHall",
100000);
HotelHall* H2 = new HotelHall(005, 4, "GoldHall",
175000);
HotelHall* H3 = new HotelHall(006, 1, "BallHall",
165000);

//GoldHall inheritance constructor
GoldHall* G1 = new GoldHall();

//SkyHall inheritance constructor
SkyHall* S1 = new SkyHall();

//BallHall inheritance constructor
BallHall* B1 = new BallHall();

cout << "-----Hotel Reservation System For
Special Events-----" << endl;

//display customer with reservation
C1->displayCustomer();
C2->displayCustomer();

cout << "-----"
<< endl;

//display hotel hall

```

```

        B1->displayhall();
        B2->displayhall();

        cout << "-----"
<< endl;

        //display receptionist
        RP1->displayDetails();
        RP2->displayDetails();

        cout << "-----"
<< endl;

        //Payment composition
        P1->displayPay();
        P2->displayPay();

        cout << "-----"
<< endl;

        //calculate discount and display discount
        D1->addDiscount(10, P1);
        D1->displayDiscount();

        cout << "-----"
<< endl;

        //add discount constructor
        D2->addDiscount(10, P2);
        D2->displayDiscount();

        cout << "-----"
<< endl;

        //calculate total
        P1->calTotalPayment();
        P2->calTotalPayment();

        return 0;
}

```

Customer.h

```
#pragma once
#include "Reservation.h"
#include "Feedback.h"
#define size 2
class Customer
{
private:
    int customerId;
    char name[20];
    char nic[12];
    char email[30];
    char address[50];
    int noOfRes;
    Reservation *res[size];
    Feedback *fdback[size];

public:
    Customer();
    Customer(int cid, const char cname[], const char
cNic[], const char cEmail[], const char cAddress[], int
nRes);
    void addReservation(Reservation*r); //As the
multiplicity 0..*
    void displayCustomer();
    void displayCustomerName();
    void validate();
    ~Customer();
};
```

Reservation.h

```
#pragma once
#include "Customer.h"
#include "Receipt.h"
#include "HotelHall.h"
```

```

class Reservation
{
private:
    char reservationId[10];
    Customer* customer;
    Receipt* Rec;
    HotelHall* hall[3];

public:
    Reservation();
    Reservation(const char rId[], Customer*cus,
Receipt*rpt);
    double calTotal();//
    void displayReservation();
    void addHall(HotelHall*hall1, HotelHall* hall2,
HotelHall* hall3);
    void displayReservationCustomer();
    void displayAvailability();
    ~Reservation();

};

```

Receptionist.h

```

#pragma once
#include "Receipt.h"
#define size 2
class Receptionist
{
private:
    int staffId;
    char staffName[20];
    char staffEmail[40];
    Receipt* rec[size];

public:
    Receptionist();

```

```

    Receptionist(int sId, const char sName[], const
char sEmail[]);
    void displayDetails();
    ~Receptionist();

};

```

Payment.h

```

#pragma once
#include "Reservation.h"
class Payment
{
private:
    int paymentId;
    char type[20];
    double totalCharge;
    char date;
    Reservation* reservation[2]; //composition

public:
    Payment();
    Payment(int pId, const char pType[], double
tCharge, const char pDate, int r1, int r2);
    double calTotalPayment();
    void displayPay();
    void validatePay();
    ~Payment();

};

```

HotelHall.h

```
#pragma once
class HotelHall
{
protected:
    int hallId;
    int package;
    char halltype[10];
    double price;

public:
    HotelHall();
    HotelHall(int hId, int pkg, const char hType[],
double hPrice);
    virtual void displayHall();
    ~HotelHall();
};
```

BallHall.h

```
#pragma once
#include"HotelHall.h"
class BallHall:HotelHall
{
private:
    int hNo;
public:
    void displayhall();
};
```

Goldhall.h

```
#pragma once
#include"HotelHall.h"
class GoldHall:HotelHall
{
private:
```



```
    int hNo;
public:
    void displayhall();

};
```

SkyHall.h

```
#pragma once
#include "HotelHall.h"
class SkyHall:HotelHall
{
private:
    int hNo;
public:
    void displayhall();
};
```

FeedBack.h

```
#pragma once
#include "Customer.h"
class Feedback
{
private:
    int feedbackId;
    Customer* customer;
    char description;

public:
    Feedback();
    Feedback(int fId, Customer* customer, const char
fDesc[]);
    void sendFeedback();
    void checkReply();
    ~Feedback();

};
```

Discount.h

```
#pragma once
#include "Payment.h"
class Discount
{
private:
    int discountId;
    double discount;

public:
    Discount();
    Discount(int dId);
    void addDiscount(double dis, Payment*
p); //dependancy(depent on class payament)
    void displayDiscount();
    ~Discount();
};
```

Receipt.h

```
#pragma once
#include "Receptionist.h"
#include "Reservation.h"
#include "Payment.h"
#define size 2
class Receipt
{
private:
    int billId;
    Receptionist* receptionist;
    Reservation* reservation;
    Payment* payment[size];
    Customer* customer[size];

public:
```

```

        Receipt();
        Receipt(int bId, Reservation*rsvt,
Receptionist*rpt);
        void displayBillDetails() {

};
        void displayCustomerDetails(Customer* c1,
Customer* c2) {
            customer[0] = c1;
            customer[1] = c2;

            for (int i = 0; i < size; i++) {
                customer[i]->displayCustomer();
            };
        };
        ~Receipt();
};

```

Customer.cpp

```

#include "Customer.h"
#include<iostream>
#include<cstring>
#include<iomanip>

using namespace std;

//constructor
Customer::Customer() {
    customerId = 0;
    strcpy_s(name, "");
    strcpy_s(nic, "");
    strcpy_s(email, "");
    strcpy_s(address, "");
    noOfRes = 0;
}
//Overloading
Customer::Customer(int cid, const char cName[], const
char cNic[], const char cEmail[], const char
cAddress[], int nRes) {

```

```

        customerId = cid;
        strcpy_s(name, cName);
        strcpy_s(nic, cNic);
        strcpy_s(email, cEmail);
        strcpy_s(address, cAddress);
        noOfRes = nRes;
    }
    void addReservation(Reservation* r) {
        if (noOfRes < size) {
            reservation[noOfRes] = r;
            noOfRes++;
        }
    }
    void displayCustomer() {
        cout << "Reservation Made : ";
        for (int i = 0; i < noOfRes; i++) {
            reservation[i]->displayReservation();
        }
    }
    void displayCustomerName() {
        cout << setw(20) << "Customer Details : " <<endl<<
        "Name : " << name << "ID : " << customerId << endl;
    }
    void validate() {

    }
    Customer::~~Customer() {
        cout << "Delete registered usewr." << endl;
    }

```

Discount.cpp

```

#include "Discount.h"
#include <iostream>
using namespace std;
Discount::Discount()
{
    discountId = 0;
    discount = 0.0;
}

```

```

}

Discount::Discount(int dId)
{
    discountId = dId;
}

void Discount::addDiscount(double dis, Payment* p)
{
    discount = (dis * p->calTotalPayment()) / 100;
}

void Discount::displayDiscount()
{
    cout << "Discount generated." << endl;
    cout << "Discount ID : " << discountId <<
    "Discount : Rs " << discount << endl;
}

Discount::~Discount()
{
}

```

Feedback.cpp

```

#include "Feedback.h"
#include<cstring>
using namespace std;
Feedback::Feedback() {
    feedbackId = 0;
    strcpy_s(description, "");
}
Feedback::Feedback(int fId, Customer* customer, const
char fDesc[]) {
    feedbackId = fId;
    strcpy_s(description, fDesc);
}void Feedback::sendFeedback() {
}

```

```
void Feedback::checkReply() {  
  
}  
Feedback::~~Feedback() {  
  
}
```

HotelHall.cpp

```
#include "HotelHall.h"  
#include<iostream>  
#include<cstring>  
using namespace std;  
  
HotelHall::HotelHall() {  
    hallId = 0;  
    package = 0;  
    strcpy_s(halltype, "");  
    price = 0.0;  
  
}  
HotelHall::HotelHall(int hId, int pkg, const char  
hType[], double hPrice) {  
    hallId = hId;  
    package = pkg;  
    strcpy_s(halltype, hType);  
    price = hPrice;  
}  
void HotelHall::displayHall() {  
  
}  
HotelHall::~~HotelHall() {  
    cout << "Delete Hall." << endl;  
}
```

BallHall.cpp

```
#include "BallHall.h"
#include<iostream>
using namespace std;
void displayhall() {
    cout << " Select Ball Hall." << endl;
}
```

GoldHall.cpp

```
#include "GoldHall.h"
#include<iostream>
using namespace std;
void displayhall() {
    cout << "Select Gold Hall." << endl;
}
```

SkyHall.cpp

```
#include "SkyHall.h"
#include<iostream>
using namespace std;
void displayhall() {
    cout << "Select Sky Hall." << endl;
}
```

Payment.cpp

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

```

Payment::Payment() {
    paymentId = 0;
    strcpy_s( type, "");
    totalCharge = 0.0;
    reservation[0] = new Reservation[006];
    reservation[1] = new Reservation[007];
}

Payment::Payment(int pId, const char pType[], double
tCharge, const char pDate, int r1, int r2) {
    paymentId = pId;
    strcpy_s(type, pType);
    totalCharge = tCharge;
    reservation[0] = new Reservation[r1];
    reservation[1] = new Reservation[r2];

    date = pDate;
}

double calTotalPayment(){
    return totalCharge;
}

void displayPay() {
    cout << "Payment for reservation : " << endl;
    for (int i = 0; i < 2; i++) {
        reservation[i]->displayReservationCustomer();
    }

    cout << endl << "Payment ID : " << paymentId <<
endl;
    cout << "Pay Date : " << date << " Payment Mode :
" << type << endl;
}

void validatePay() {
}

Payment::~~Payment() {
    cout << "Delete payment made." << endl;
}

```


Receipt.cpp

```
#include "Receipt.h"
Receipt::Receipt() {
    billId = 0;
}
Receipt::Receipt(int bId, Reservation* rsvt,
Receptionist* rpt) {
    billId = bId;
    reservation = rsvt;
    receptionist = rpt;
}
void Receipt::displayBillDetails() {

}
Receipt::~Receipt() {

}
```

Reservation.cpp

```
#include "Reservation.h"
#include<cstring>
#include<iostream>
#include<iomanip>

using namespace std;

Reservation::Reservation() {
    strcpy_s(reservationId, "");
}
Reservation::Reservation(const char rId[], Customer*
cus, Receipt* rpt) {
    strcpy_s(reservationId, rId);
    customer = cus;
    Rec = rpt;
}
```

```

        cus->addReservation(this);
    }
    double Reservation::calTotal() {

    }
    void Reservation::addHall(HotelHall* hall1, HotelHall*
    hall2, HotelHall* hall3)
    {
        hall[0] = hall1;
        hall[1] = hall2;
        hall[2] = hall3;
    }
    //
    void Reservation:: displayReservation() {
        cout << endl << setw(20) << "Reservations : " <<
        setw(10) << "Reservation ID :" << reservationId <<
        endl;
        for (int i = 0; i < 3; i++) {
            hall[i]->displayHall();
        }
    }
    void Reservation::displayReservationCustomer() {
        cus->displayCustomerName();
    }
    void Reservation::displayAvailability() {

    }
    Reservation::~Reservation() {
        cout << "Delete Reservation." << endl;
    }

```

Receptionist.cpp

```

#include "Receptionist.h"
#include <cstring>
#include <iostream>
#include <iomanip>
using namespace std;
Receptionist::Receptionist() {

```

```

        staffId = 0;
        strcpy_s(staffName, "");
        strcpy_s(staffEmail, "");
    }
    Receptionist::Receptionist(int sId, const char sName[],
    const char sEmail[]) {
        staffId = sId;
        strcpy_s(staffName, sName);
        strcpy_s(staffEmail, sEmail);
    }
    void Receptionist::displayDetails() {
        cout << setw(20)<<"Receptionist Details " <<
        setw(20)<<"Receptionist ID : "<<staffId <<"Receptionist
        Name : "<<staffName<<endl;
        cout << "Receptionist Email : " << staffEmail <<
        endl;
    }
    Receptionist::~~Receptionist() {
}

```

Screenshots of the executed program

```
Microsoft Visual Studio Debug Console
-----Hotel Reservation System For Special Events-----
Customer Details :
Name : Ruslth           ID : 001
Reservations           Reservation ID : 001
Name : Hansaka          ID : 002
Reservations           Reservation ID : 002
-----
Select Ball Hall
Select Gold Hall
-----
Receptionist
Receptionist Details  Receptionist ID : 006 Receptionist Name : Amanda
Receptionist Email : amanda123@gmail.com
Receptionist Details  Receptionist ID : 007 Receptionist Name : Menula
Receptionist Email : menula123@gmail.com
-----
Payment ID : 001
Pay Date : 2022-06-02      Payment Mode : Credit Card
Payment ID : 002
Pay Date : 2022-07-22      Payment Mode : Debit Card
-----
Discount generated
Discount ID : 001          Discount : Rs 10000
Discount generated
Discount ID : 002          Discount : Rs 17500
-----
Total Payment : Rs 90000
Total Payment : Rs 157500

C:\Users\Hansaka\source\repos\HotelReservationSystemAssignment\x64\Debug\HotelReservationSystemAssignment.exe (process 9536) exited with code 0.
Press any key to close this window . . .
```