

Sri Lanka Institute of Information Technology



Assignment

MLB_WE_01.01_04

Hotel Reservation System for Weddings

Object Oriented Concepts – IT1050

B.Sc. (Hons) in Information Technology

Group Details

Group Project ID : MLB_WE_01.01_04

Project Title : **Hotel Reservation System for Weddings**

	Student ID	Student Name	Email	Contact Number
1	IT21281150	W.M.L.K.Wickramasinghe	it21281150@my.sliit.lk	0789736535
2	IT21273780	R.M.P.C Premathilake	it21273780@my.sliit.lk	0703877440
3	IT21316104	A A Bary	it21316104@my.sliit.lk	0775599733
4	IT21340864	E M N Edirisinghe	it21340864@my.sliit.lk	0752324098
5	IT21294020	A P P G Abeysinghe	it21294020@my.sliit.lk	0761029167

System Requirements

- Guests can view the system and must register on the system by entering personal details to make a hotel reservation.
- Registered users can log in to the system after entering the correct username and password in their account.
- Registered users can use the system to book a hotel, wedding hall, wedding package, food package and pay for it online.
- When a user enters their personal information and registers in the system, a customer id unique to the user is automatically created.
- The system automatically issues a booking id and payment id once a user has paid for a booking.
- The system will automatically record the date of payment and the date of booking the hotel and wedding hall after making a reservation payment.
- When making a payment, the payment type and card details should be included.
- After the customer has made the payment and the bank has confirmed the payment, a message should be sent to inform the customer.
- The system should be operational 24 hours a day, every day of the year
- system admin can change the details of hotels, wedding halls, wedding packages, food packages on the website.

Classes

- Customer
- Registered customer
- Booking
- Payment
- Offer
- Employee
- Wedding package
- Food package
- Hotel

Methods

- Customer - Register to the system
- Registered customer – Login to the system
 - Choose wedding package
 - Choose food package
 - Choose hotel
 - Choose hall
 - Do payment
- Booking -
 - Generate booking ID
 - Check available hotels
 - Check available halls
 - Show available wedding packages
 - Show available food packages
- Payment –
 - Generate pay ID
 - Calculate fees
 - Check transaction type
 - Check payment detail
 - Confirm payments
- Offer -
 - Generate offer id
 - Add new seasonal offer
- Employee -
 - login the system
 - Confirm bookings
 - Confirm offers
- Wedding package -
 - Generate wedding package code
 - Add wedding package detail
 - Delete and update wedding package detail
- Food package -
 - Generate food package code
 - Add food package detail
 - Delete and update food package detail
- Hotel -
 - Generate hotel ID
 - Add hotels details and location
 - Delete and update hotel details and location

CRC Cards

Customer	
Responsibility	Collaborators
Register to the System	
Allow to View Hotels, Halls, Wedding Packages, Food Packages	Hotels Wedding Packages, Food Packages

Registered Customer	
Responsibility	Collaborators
Allow to select Hotels, Halls, Wedding Packages, Food Packages	Booking
Add and update customer details	

Booking	
Responsibility	Collaborators
Place a Booking	Hotel, Wedding package, Food Package
Check availability of Hotels and Wedding Halls	Hotel
Calculate the Total price	Payment

Payment	
Responsibility	Collaborators
Make a new payment	
Generate Payment ID	
Check payment details	Employee
Confirm payment details	Employee

Offers	
Responsibility	Collaborators
Check available offers	
Add new offers	System Admin
Delete offers	System Admin

Employee	
Responsibility	Collaborators
Login to the system	
Send Booking details to staff members	Booking
Send Booking details report to Manager	Booking
Check Booking Details	Booking

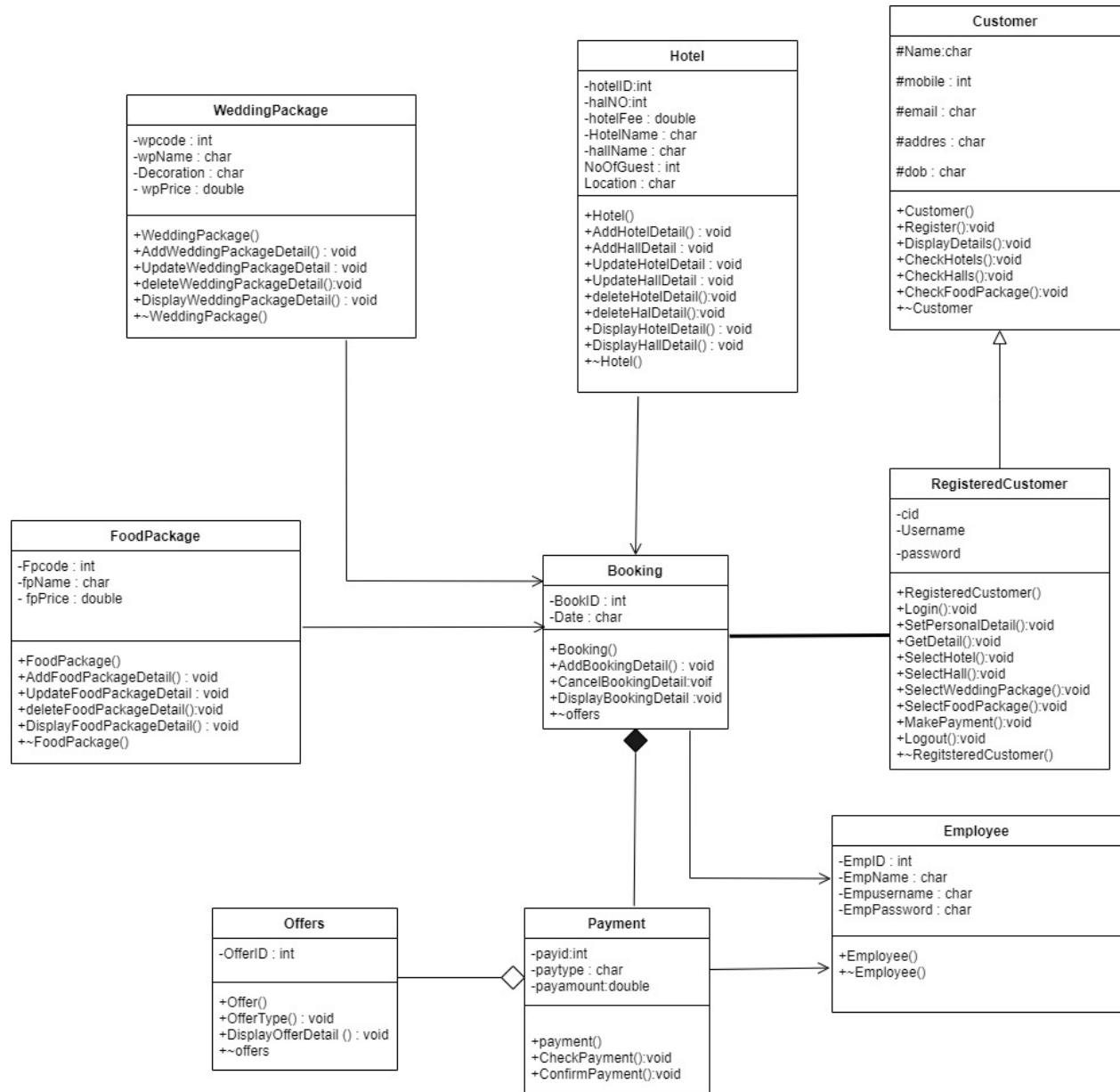
Wedding Package	
Responsibility	Collaborators
Add Wedding Package Details	System Admin
Delete Wedding Package Details	System Admin
Update Wedding Package Details	System Admin

Food Package	
Responsibility	Collaborators
Add Food Package Details	System Admin
Delete Food Package Details	System Admin
Update Food Package Details	System Admin

Hotel	
Responsibility	Collaborators
Add Hotel Details	System Admin
Delete Hotel Details	System Admin
Update Hotel Details	System Admin

Report	
Responsibility	Collaborators
Generate Booking details	Booking
Generate Payment details	Payment

Class Diagram (UML Notation)



weddingPackage.cpp

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

weddingPackage::weddingPackage()
{

}

weddingPackage::weddingPackage(int wcode, const char wName[], double wPrice, const
char wDecorations[])
{
    wpcode = wcode;
    strcpy(Decorations, wDecorations);
    wpPrice = wPrice;
    strcpy(wpName, wName);
}

void AddweddingPackageDetails(int wcode, const char wName[], double wPrice, const
char wDecorations[])
{

}

void weddingPackage::updateWeddingPackageDetails()
{

}

void weddingPackage::DeleteWeddingPackageDetails()
{

}

void weddingPackage::DisplayWeddingPackageDetails()
{
    cout << "Wedding Package Code : " << wpcode << endl
        << "Wedding Package Name : " << wpName << endl
        << "Decoration : " << Decorations << endl
        << "Wedding Package Price : " << wpPrice << endl;
}

weddingPackage::~weddingPackage()
{

}
```

weddingPackage.h

```
#include<cstring>

using namespace std;

class weddingPackage
{
private:
    int wpcode;
    char wpName[10];
    char Decorations[30];
    double wpPrice;

public:
    weddingPackage();
    weddingPackage(int wcode, const char wName[], double wPrice, const char
wDecorations[]);
    void AddweddingPackageDetails(int wcode, const char wName[], double wPrice,
const char wDecorations[]);
    void updateWeddingPackageDetails();
    void DeleteWeddingPackageDetails();
    void DisplayWeddingPackageDetails();
    ~weddingPackage();

};
```

FoodPackage.cpp

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

FoodPackage::FoodPackage()
{
}

FoodPackage::FoodPackage(int Fcode, const char FName[], double FPrice)
{
    FPCode = Fcode;
    strcpy(FPName, FName);
    FPPrice = FPrice;
}

void FoodPackage::AddFoodPackageDetails(int Fcode, const char FName[], double
FPrice)
{
}

void FoodPackage::updateFoodPackageDetails()
{
}

void FoodPackage::DeleteFoodPackageDetails()
{
}

void FoodPackage::DisplayFoodPackageDetails()
{
    cout << "Food Package Code : " << FPCode << endl
         << "Food Package Name : " << FPName << endl
         << "Food Package Price : " << FPPrice << endl;
}

FoodPackage::~~FoodPackage()
{
}
```

FoodPackage.h

```
#include<cstring>

using namespace std;

class FoodPackage
{
private:
    int FPcode;
    char FPName[10];
    double FPPrice;

public:
    FoodPackage();
    FoodPackage(int Fcode, const char FName[], double FPrice);
    void AddFoodPackageDetails(int Fcode, const char FName[], double FPrice);
    void updateFoodPackageDetails();
    void DeleteFoodPackageDetails();
    void DisplayFoodPackageDetails();
    ~FoodPackage();

};
```

Customer.cpp

```
#include "Customer.h"
```

```
#include <cstring>
```

```
#include <iostream>
```

```
using namespace std;
```

```
Customer::Customer()
```

```
{  
}
```

```
Customer::Customer(const char cname[], int cmobile, const char cemail[], const char caddres[], const char cdob[])
```

```
{
```

```
    strcpy(Name, cname);
```

```
    strcpy(email, cemail);
```

```
    mobile = cmobile;
```

```
    strcpy(addres, caddres);
```

```
    strcpy(dob, cdob);
```

```
}
```

```
void Customer::Register()
```

```
{  
}
```

```
void Customer::DisplayDetail()
```

```
{
```

```
        cout<<"Customer Detail"<<endl
            <<"-----"<<endl
            << "Customer Name : " << Name << endl
            << "Email : " << email << endl
            << "Mobile No : " << mobile << endl
            << "addres : " << addres << endl
            << "Date Of Birth" << dob << endl<<endl;
    }
```

```
void Customer::CheckHotels()
{
}
```

```
void Customer::CheckHalls()
{
}
```

```
void Customer::CheckWeddingPackages()
{
}
```

```
void Customer::CheckFoodPackages()
{
}
```

```
Customer::~~Customer()
{
}
```

Customer.h

```
class Customer
```

```
{
```

```
protected:
```

```
    char Name[20];
```

```
    int mobile;
```

```
    char email[15];
```

```
    char addres[20];
```

```
    char dob[15];
```

```
public:
```

```
    Customer();
```

```
    Customer(const char cname[], int cmobile,const char cemail[],const char caddres[],const char  
cdob[]);
```

```
    void Register();
```

```
    void DisplayDetail();
```

```
    void CheckHotels();
```

```
    void CheckHalls();
```

```
    void CheckWeddingPackages();
```

```
    void CheckFoodPackages();
```

```
    ~Customer();
```

```
};
```


RegisterdCustomer.cpp

```
#pragma warning(disable:4996)
```

```
#include "Customer.h"
```

```
#include "RegisteredCustomer.h"
```

```
#include <cstring>
```

```
#include <iostream>
```

```
using namespace std;
```

```
RegisteredCustomer::RegisteredCustomer()
```

```
{
```

```
    cid = 0;
```

```
    strcpy(username, "");
```

```
}
```

```
RegisteredCustomer::RegisteredCustomer(int custid, const char  
username[]):Customer(Name,mobile,email,addres,dob)
```

```
{
```

```
    cid = custid;
```

```
    strcpy(username, cusername);
```

```
}
```

```
void RegisteredCustomer::setPersonalDetails(const char cname[],const char cemail[], int cmobile, const  
char caddres[], const char cdob[])
```

```
{
```

```
    strcpy(Name, cname);
```

```
    strcpy(email, cemail);
```

```
    mobile = cmobile;
```

```

        strcpy(dob, cdob);
        strcpy(address, caddress);
    }

void RegisteredCustomer::DisplayDetails()
{
    cout << "Registered Customer Detail" << endl
           << "-----" << endl
    << "Registered Customer ID : " << cid << endl
           << "Username : " << username << endl
           << "Name : " << Name << endl
           << "Email : " << email << endl
           << "Mobile No:" << mobile << endl
           << "Address : " << address << endl
           << "Date Of Birth : " << dob << endl << endl;
}

void RegisteredCustomer::SelectHotel()
{
}

void RegisteredCustomer::SelectHall()
{
}

void RegisteredCustomer::SelectWeddingPackage()
{
}

```

```
void RegisteredCustomer::SelectFoodPackage()
{
}
```

```
void RegisteredCustomer::MakePayment()
{
}
```

```
void RegisteredCustomer::logout()
{
}
```

```
RegisteredCustomer::~~RegisteredCustomer()
{
}
```

RegisterdCustomer.h

```
class RegisteredCustomer : public Customer
{
private:
    int cid;
    char username[10];
    char password[8];

public:
    RegisteredCustomer();
    RegisteredCustomer(int custid, const char cusername[]);
    void setPersonalDetails(const char cname[],const char cemail[], int cmobile, const char
caddres[], const char cdob[]);
    void DisplayDetails();
    void SelectHotel();
    void SelectHall();
    void SelectWeddingPackage();
    void SelectFoodPackage();
    void MakePayment();
    void logout();
    ~RegisteredCustomer();
};
```

Payment.cpp

```
#include <iostream>
```

```
#include "payment.h"
```

```
#include <cstring>
```

```
using namespace std;
```

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

```
{
```

```
    payID = 0;
```

```
    strcpy(payType, "");
```

```
    payAmount = 0;
```

```
}
```

```
void Payment :: setPayment(int pID , const char ppayType[], double ppayAmount)
```

```
{
```

```
    payID = pID;
```

```
    strcpy(payType , ppayType);
```

```
    payAmount = ppayAmount;
```

```
}
```

```
void Payment :: checkPayment()
```

```
{
```

```
}
```

```
void Payment :: confirmPayment()
```

```
{  
}
```

```
void Payment :: displayPaymentDetails()
```

```
{  
    cout << "Payment Details" << endl;  
    cout << "-----" << endl;  
    cout << "payID : " << payID << endl;  
    cout << "Payment Type : " << payType << endl;  
    cout << "Payment Amount : " << payAmount << endl;  
    cout << "-----" << endl;  
}
```

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

```
{  
  
    cout << "Delete Payment Class" << endl;  
}
```

Payment.h

```
class Payment
{
    private:
        int payID;
        char payType[20];
        double payAmount;

    public:
        Payment();
        void setPayment(int pID , const char ppayType[] , double ppayAmount);
        void checkPayment();
        void confirmPayment();
        void displayPaymentDetails();
        ~Payment();
};
```

Offers.cpp

```
#include <iostream>
```

```
#include "offers.h"
```

```
#include <cstring>
```

```
using namespace std;
```

```
offers::offers()
```

```
{
```

```
    offerID = 0;
```

```
}
```

```
offers::offers(int oID)
```

```
{
```

```
    offerID = oID;
```

```
}
```

```
void offers::offertype()
```

```
{
```

```
}
```

```
void offers::displayOfferDetails()
```

```
{
```



```
    cout << "Offers" << endl  
    << "-----" << endl  
    << "offerID : " << offerID << endl  
    << "-----" << endl << endl;  
}
```

```
offers::~~offers()  
{  
    cout << "Delete Payment Class " << endl;  
}
```

Offers.h

class offers

{

private:

int offerID;

public:

offers();

offers(int oID);

void offertype();

void displayOfferDetails();

~offers();

};

Main.cpp

```
#include "Customer.h"
#include "RegisteredCustomer.h"
#include "weddingPackage.h"
#include "FoodPackage.h"
#include "payment.h"
#include "offers.h"
#include "booking.h"
#include <iostream>
#include <cstring>
using namespace std;

int main()

{
    //customer
    Customer* c1;

    c1 = new Customer("Ranajan",776585487,"ranjan@gmail.com","Kesbew","1996-06-23");

    c1->DisplayDetail();
```

```
//Registered customer
RegisteredCustomer* rc1;

rc1 = new RegisteredCustomer(01, "SamanP");

rc1->setPersonalDetails("Saman Perera","saman@gmail.com", 778596489, "Dehiwala", "1986-05-25");

rc1->DisplayDetails();


//Wedding package
weddingPackage* wp1;

wp1 = new weddingPackage(333001,"classic", 800000,"white and blue theme");

wp1->DisplayWeddingPackageDetails();


//Food Package
FoodPackage* fp1;

fp1 = new FoodPackage(989001, "Silver", 2500);

fp1->DisplayFoodPackageDetails();
```

```
//Payment
```

```
Payment* py1;
```

```
py1 = new Payment();
```

```
py1->setPayment(21, "Cash", 1000);
```

```
py1->displayPaymentDetails();
```

```
//Offers
```

```
offers* of1;
```

```
of1 = new offers(001);
```

```
of1->displayOfferDetails();
```

```
//booking
```

```
booking* book1;
```

```
book1 = new booking(100, 2, "SamanP", "2022-05-21");
```

```
book1->setbookingDetail(8801, 05, 4532, 1245);
```

```
book1->DisplayBookingDetail();
```

```
delete c1,rc1,wp1,fp1,of1,book1;
```

```
return 0;
```

```
}
```

booking.h

```
class booking
{
private:
    int bookID;
    char date[10];
    int cid;
    char username[8];
    int hotelID;
    int hallNO;
    int WpCode;
    int Fpcode;

public:
    booking();

    booking(int cbookID, int bcid , const char cusername[], const char cdate[]);
    void setbookingDetail(int chotelID, int ChallID, int cWpCode, int FpCode);
    void cancelBooking();
    void DisplayBookingDetail();
    ~booking();
};
```

Booking.cpp

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

booking::booking()
{

}

booking::booking(int cbookID, int bcid, const char cusername[], const char cdate[])
{
```

```

        bookID = cbookID;
        cid = bcid;
        strcpy(username, cusername);
        strcpy(date, cdate);
    }

void booking::setbookingDetail(int chotelID, int ChallID, int cWpCode, int cFpCode)
{
    chotelID = hotelID;
    ChallID = hallNO;
    cWpCode = WpCode;
    cFpCode = Fpcode;
}

void booking::cancelBooking()
{
}

void booking::DisplayBookingDetail()
{
    cout << "Booking Details" << endl
        << "-----" << endl
        << "Booking ID : " << bookID << endl
        << "Customer ID : " << cid << endl
        << "Customer username : " << username << endl
        << "Date : " << date << endl
        << "Hotel ID : " << hotelID << endl
        << "Hall ID : " << hallNO << endl
        << "Wedding Package code : " << WpCode << endl
        << "Food package code : " << Fpcode << endl << endl;
}

booking::~~booking()
{
}

```



```
Microsoft Visual Studio Debug Console
Customer Detail
-----
Customer Name : Ranajan
Email : ranjan@gmail.com
Mobile No : 776585487
Address : Kenilow
Date Of Birth : 1996-06-23

Registered Customer Detail
-----
Registered Customer ID : 1
Username : SamanP
Name : Saman Perera
Email : saman@gmail.com
Mobile No : 776585487
Address : Dehiwala
Date Of Birth : 1980-05-25

Wedding Package Code : 333001
Wedding Package Name : classic
Decoration : white and blue theme
Wedding Package Price : 800000

Food Package Code : 989001
Food Package Name : Silver
Food Package Price : 2500

Payment Details
-----
payID : 21
Payment Type : Cash
Payment Amount : 1500
-----

Offers
-----
offerID : 1
-----

Booking Details
-----
Booking ID : 100
Customer ID : 2
Customer username : SamanP
Date : 2022-05-21
Hotel ID : -842150451
Hall ID : -842150451
Wedding Package code : -842150451
Food package code : -842150451

D:\SLT\IT\OC\Assignment\ConsoleApplication1\vs4\Debug\ConsoleApplication1.exe (process 19472) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Individual Contribution

	Student ID	Student Name	Individual Contribution
1	IT21281150	W.M.L.K.Wickramasinghe	<ul style="list-style-type: none">• System Requirements• weddingPackage.cpp• weddingPackage.h• FoodPackage.cpp• FoodPackage.h
2	IT21273780	R.M.P.C Premathilake	<ul style="list-style-type: none">• Hotel.cpp• Hotel.h• Booking.cpp• Booking.h
3	IT21316104	A A Bary	<ul style="list-style-type: none">• CRC Cards• Employee.cpp• Employee.h
4	IT21340864	E M N Edirisinghe	<ul style="list-style-type: none">• Classes, Methods• Customer.cpp• RegisterdCustomer.cpp• Customer.h• RegisterdCustomer.h
5	IT21294020	A P P G Abeysinghe	<ul style="list-style-type: none">• Payment.cpp• Payment.h• Offers.cpp• Offers.h