Topic : **Hotel Reservation System For Weddings**

Group no : **MLB_PG.04.02_12**

Campus : Malabe

Submission Date :

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.

Exercise 1:

Registration No	Name	Contact Number
IT20120184	M.K.C.N. Mandawela	0715226008
IT20116224	W.R.V.K. Thathsarani	0775832200
IT20466312	J.S.T. Silva	0768422296
IT20260088	N.K.T.S. Naragala	0750941492
IT20163990	N.P. Maduranga	0772436443

1) The following are the user requirements of the Hotel Reservation System.

- Customer go to our official website, viewing gallery and providing details such as name, email, address and mobile number.
- Customer enter the Username & password for website login.
- Customer register to the website and search hotel for wedding.
- Customer checks the availability of the selected hotel.
- The customer uses different payment methods for each packages such as PayPal, credit card, debit card, visa card & Cheque.
- Customer confirms the hotel booking and the payment are validated.
- Customer gives feedbacks on hall and services.
- Chef checks the kitchen food store and order high quality fresh food items.
- Chef makes the food menu for weddings and functions.
- Receptionist checks the hotel check in and checkout time.
- Decoration Manager checks for new decoration packages and order new decoration items.
- Hotel Manager adds new features in the site such as give new discount, give new rules, new conditions & add new titles.
- Web staff visit the notification gallery; system shows customer feedbacks to web staff.
- Web staff checks feedbacks and get a record of customer feedbacks.
- Web staff monitor database. Such as customer name, customer orders, customer email address and mobile number & customer problems.
- Web staff update the stock database such as modify items, delete items and items.
- System administrator chooses user payments, system shows the details & Cheque number about user payments to web owner such as PayPal, credit card & Debit card.
- Owner accept the payment.

2) Nouns

- Hotel reservation system
- Customer
- Website
- Name
- Address
- Email
- Mobile number
- Username
- Password
- Hotel
- Decoration Manager
- Packages
- PayPal
- Credit Card
- Debit Card
- Cheque
- Hall
- Service
- Food items
- Food store
- Food menu
- Functions
- Receptionist
- Wedding
- Decoration items
- features
- Discount
- Rules
- Conditions
- Titles
- Chef
- Administrator

- Details
- Notification Gallery
- Web staff
- Feedbacks
- Booking
- Order
- Customer Name
- Customer Orders
- Customer address
- Customer mobile number
- Customer Problems
- Items
- Owner
- Payment
- Cheque number

3) List down the classes

Hotel Reservation System	
Customer	
Name	Attribute
Address	Attribute
Email	Out of scope
Phone Number	Attribute
Website	Attribute
Username	Attribute
Password	Attribute
Order	
Food items	Attribute
Food menu	Attribute
Details	Meta language
Item	
Decoration Manager	Redundant
Chef	Redundant

Receptionist	Redundant
Food store	Redundant
Booking	
Hotel	Attribute
Services	Attribute
Hall	Attribute
Functions	Attribute
Wedding	Attribute
Decoration items	Attribute
Payment	
Customer name	Attribute
Owner	Redundant
Administrator	Redundant
Features	
Notification Gallery	Attribute
Package	Attribute
Web staff	Redundant
Feedbacks	Attribute
Customer problems	Attribute
Title	Attribute
Discount	Attribute
Rules	Attribute
Conditions	Attribute
Cheque	
Check number	Attribute
Customer order	Attribute
Customer address	Attribute
Customer mobile number	Attribute
Credit	
PayPal	Out of scope
Credit card	Out of scope
Debit card	Out of scope

Classes: Customer, Order, Item, Payment, Features, Booking, Cheque, Credit

Exercise 2:

Class Name: **Customer**	
Responsibities	Collaborations
Register to the website	
Enter details	
Search hotels	
Confirms the booking	Booking
Making Payment	Payment

Class Name: **Items**	
Responsibities	Collaborations
Modify items	
Delete items	
Add items	

Class Name: **Features**	
Responsibities	Collaborations
Give new discount	Payment
Give new rules	
New conditions	
Add new titles	

Class Name: **Payment**	
Responsibities	Collaborations
Accept the payment	Customer
Add discount	
System shown the details about user payment	Customer

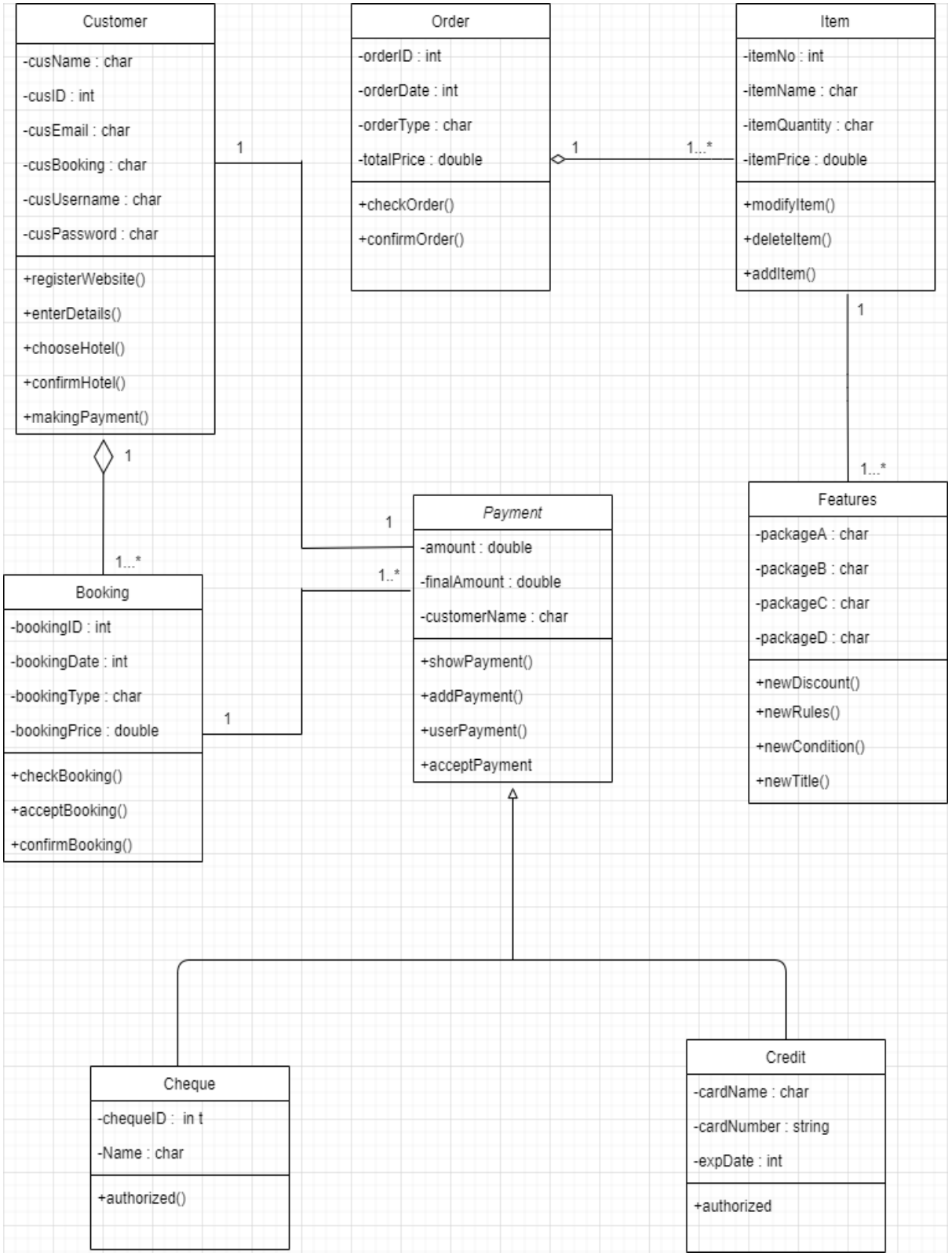
Class Name: **Order**	
Responsibities	Collaborations
Check the new orders	
Accept orders	
Confirms the order	

Class Name: **Credit**	
Responsibities	Collaborations
Check the expDate	

Class Name: **Cheque**	
Responsibities	Collaborations
Check the cheque details	
Check the final Amount	

Class Name: **Booking**	
Responsibities	Collaborations
Registered on the website	
Chooses the hall & package	Customer
Selected the date	

Exercise 3:


Customer
-cusName : char
-cusID : int
-cusEmail : char
-cusBooking : char
-cusUsername : char
-cusPassword : char
+registerWebsite()
+enterDetails()
+chooseHotel()
+confirmHotel()
+makingPayment()
Order
-orderID : int
-orderDate : int
-orderType : char
-totalPrice : double
+checkOrder()
+confirmOrder()
Item
-itemNo : int
-itemName : char
-itemQuantity : char
-itemPrice : double
+modifyItem()
+deleteItem()
+addItem()
Booking
-bookingID : int
-bookingDate : int
-bookingType : char
-bookingPrice : double
+checkBooking()
+acceptBooking()
+confirmBooking()
Payment
-amount : double
-finalAmount : double
-customerName : char
+showPayment()
+addPayment()
+userPayment()
+acceptPayment
Features
-packageA : char
-packageB : char
-packageC : char
-packageD : char
+newDiscount()
+newRules()
+newCondition()
+newTitle()
Cheque
-chequeID : in t
-Name : char
+authorized()
Credit
-cardName : char
-cardNumber : string
-expDate : int
+authorized
1
1...*
1..*

Exercise 4:

Main.cpp

```
#include <iostream>

#include <cstring>

#include "Customer.h"

#include "Features.h"

#include "Item.h"

#include "Booking.h"

#include "Order.h"

#include "Payment.h"

using namespace std;


int main() {

Customer* cus1;

cus1 = new Customer("Kevin Perera", "C110", "kevins98@gmail.com", "PackageB", "KevinS"
, "Kevin1234");

Features* fet1;

fet1 = new Features("Package A", "Package B", "Package C","Package D");

Item* ite1;

ite1 = new Item("IT185", "Poru Items", "2", "15 000.00");

Booking* book1;

book1 = new Booking("BK102","02/11/2021","Indoor Wedding Hall","90 000.00");

Order* ord1;

ord1 = new Order("OD109", "01/31/2021", "OD Type 3", "25 000.00");

Payment* pay1;

pay1 = new Payment ("Kevin Perera", "90 000.00", "150 000.00 ");

}
```

Customer.h

```
#pragma once

class Customer

{

	private:

		int cusId;

		int cusMobileno[10];

		char cusName[20];

		char cusEmail[20];

		char cusBooking[20];

		char cusUsername[20];

		char cusPassword[20];

		Payment* payment[10];

		 Booking* booking[10];

	public:

	Customer(int csId, char csName[20], char csEmail[20], char csBooking[20], char c
	sUsername[20], char csPassword[20]);

	void registerWebsite();

	void enterDetails();

	void chooseHotel();

	void confirmHotel();

	void makingPayment();

	~Customer();

};
```

Customer.cpp

```
//Implementation part

#include <iostream>

#include"Customer.h"

#include<cstring>

using namspace std;

Customer::Customer(int csId,int csMobileno[10], char csName[20], char csEmail[20], cha
r csBooking[20], char csUsername[20], char csPassword[20])

{

cusId = csId;

cusMobileno = csMobileno;

strcpy(cusName,csName);

strcpy(cusEmail, csEmail);

strcpy(cusBooking, csBooking);

strcpy(cusUsername, csUsername);

strcpy(cusPassword, csPassword);

}

void Customer::registerWebsite()

{

}

void Customer::enterDetails()

{

}

void Customer::chooseHotel()

{

}

void Customer::confirmHotel()

{

}
```

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

Order.h

```
#program once
class Orders
{
	private:
		Items* itm[SIZE];
		int orderID;
		char orderDate[10];
		char orderTYpe[20];
		dou blePrice;
	public:
		Order();
		Order(int pOrderID, const char pOrderDate[], char pOredrType[], Customer *
		pCus);
		void chechOrder();
		void confirmOrder()
};
```

Order.cpp

```
//Implementation part

#include "Order.h"

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std;

Order::Order() {

orderID = 0;

strcpy_s(orderDate, "");

strcpy_s(orderType, "");

totalPrice = 0;

}

Order::order(int pOrderID, const char pOrderDate[], char pOredrType[], double pTotalPr
ice) {

 orderID = pOrderID;

strcpy_s(orderDate, pOrderDate);

strcpy_s(orderType, pOrderType);

totalPrice = pTotalPrice;

}
```

Item.h

```
#program once

#define SIZE 2

class Item {

      private:

            int itemNo;

            char itemName[15];
```

```
            char itemQuantity[10];
            double itemPrice;
      public:
            Item();
            Item(int pItemNo, const char pItemName[], const char pItemQuantity, double
             pPrice);
            void modifyItem;
            void deleteItem;
            void addItem;
};
```

Item.cpp

```
#include "Item.h"
#include <iostream>
#include <iomanip>
#include <cstring>
using namespace std;
Item::Item()
{
itemNo = 0;
strcpy_s(itemName, "");
strcpy_s(itemQuantity, "");
itemPrice = 0;
}
Item::Item (int pItemNo, const char pItemName[], const char pItemQuantity[], double pI
temPrice)
{
itemNo = pItemNo;
strcpy_s(itemName, pItemName);
```

```
strcpy_s(iyemQuantity, pItemQuantity);
itePrice = pItemPrice;
}
```

Booking.h

```
#pragma once
class Booking
{
	private:
		int bookingId;
		int bookingDate;
		char bookingType;
		double bookingPrice;
		Customer* customer[10];
	public:
		Booking(int bId, int bDate, char bType, double tPrice);
		void checkBooking();
		void acceptBooking();
		void confirmBooking();~Booking();
};
```

Booking.cpp

```
//Implementation part
#include<iostream>
#include"Booking.h"
#include<cstring>
using namespace std;
Booking::Booking(int bId, int bDate, char bType, double bPrice)
{
	bookingId = bId;
	bookingDate = bDate;
	strcpy(bookingType, bType);
	bookingPrice = bPrice;
}
void Booking::checkBooking()
{
}
void Booking::acceptBooking()
{
}
void Booking::confirmBooking()
{
}
Booking::~Booking()
{
}
```

Payment.h

```
#pragma once
class Payment
{
	protected:
		char cusName[20];
		double amount;
		double finalAmount;
		Customer* customer[10];
		Booking* booking[10];
	public:
		Payment(char csName[20], double pAmount, double fAmount);
		void acceptPayment();
		void showPayment();
		void addPayment();
		void userPayment();
		~Payment();
};
class Cheque:public Payment{
	private:
		char name[10];
		int chequeID;
	public:
		void authorized();
};
class Credit::public Payment {
	private:
		char cardName[10];
		int cardNumber;
```

```
            int expDate;
      public:
            void authorized();
};
```

Payment.cpp

```
//Implementation part
#include<iostream>
#include<cstring>
#include"Payment.h"
using namespace std;
Payment::Payment(char csName[20], double pAmount, double fAmount)
{
      strcpy(cusName, csName);
      amount = pAmount;
      finalAmount = fAmount;
}
void Payment::acceptPayment()
{
}
void Payment::showPayment()
{
}
void Payment::addPayment()
{
}
void Payment::userPayment()
{
}
Payment::~Payment()
```

```
{
}
void Cheque::authorized()
{
}
void Credit::authorized()
{
}
```

Features.h

```
#pragma once
class Features
{
	private:
		char packageA[20];
		char packageB[20];
		char packageC[20];
		char packageD[20];
		Item* item[10];
	 public:
		Features(char packA[20], char packB[20], char packC[20], char packD[20]);
		void newDiscount();
		void newRules();
		void newConditions();
		void newTitles();
		~Features();
};
```

Features.cpp

```
//Implementation part
#include<iostream>
#include<cstring>
#include"Features.h"
using namespace std;
Features::Features(char packA[20], char packB[20], char packC[20], char packD[20])
{
	strcpy(packageA, packA);
	strcpy(packageB, packB);
	strcpy(packageC, packC);
	strcpy(packageD, packD);
}
void Features::newDiscount()
{
}
void Features::newRules()
{
}
void Features::newConditions()
{
}
void Features::newTitles()
{
}
Features::~Features()
{
}
```