Topic: Online cab and taxi reservation system

Group no: KDY_01

Campus: Kandy

Submission Date: 2022-05-16

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
IT21239434	P.H.K.C. Punchihewa	071-9640691
IT21258794	Karunasena H.G.M.K.K.L	076-6974709
IT21373770	Dissanayake D.M.U.I	077-1413244
IT21053696	Wickramasekara R.K.P	071-4077402
IT21209352	D.M.K.C. Bandara	076-4466802

1. User Requirement

1. A guest can create an account as a customer providing his email, name, phone number, address.
2. The registered customer must enter the correct username and password to login the system. After that the system check password is valid or not finally customer logout the system.
3. If unregistered customers want to use the system, they must register with the system by providing the correct details.
4. Both registered and unregistered customers can visit FAQ, contact us, about us.
5. User can change the password.
6. User can book a taxi by add relevant details like destination, start place, time, vehicle type.
7. User can cancel a booking.
8. User can get offer that generate by the system.
9. Admin can add offers using offer id, offer description to a booking.
10. User can pay the total fee in payment gateways.
11. Payment can be paid by visa, credit card, pay pal.
12. After the payment 'payment id' is generated for a booking id.
13. Customer can check payment details and confirmation.

2. Classes

Registered customer - login, check, logout, change

Unregistered customer – use, register

Booking – add, generate, cancel

Taxi – book

Payment - pay, generate, check

Offer – add

3. CRC Cards

Registered user	
Responsibility	**Collaboration**
Login to the system	
Book a taxi	Booking, taxi
Check the payment detail	payment
Change password	
Log out the system	

Unregistered user	
Responsibility	**Collaboration**
Create account	
visit tabs	

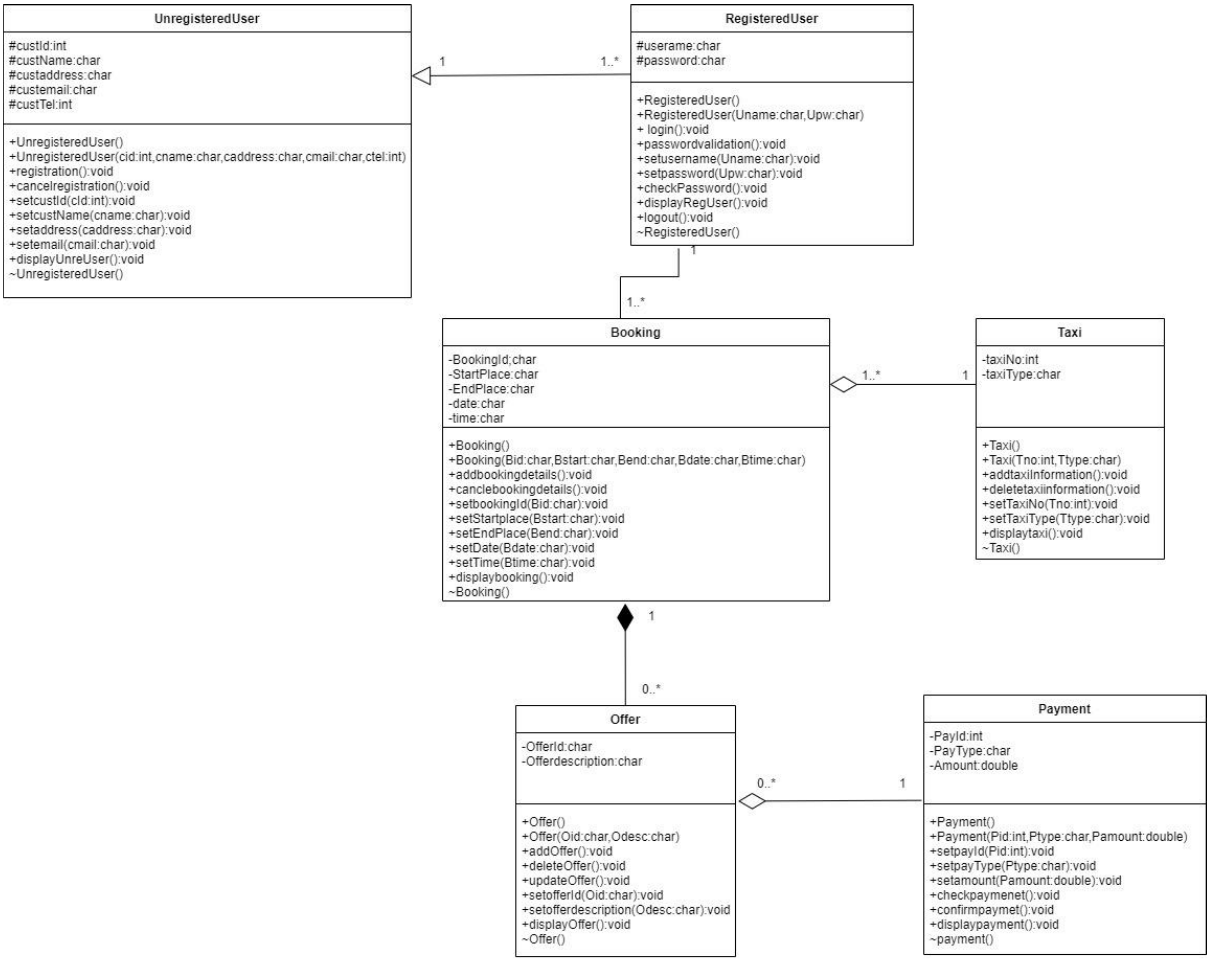
Booking	
Responsibility	**Collaboration**
Place the booking	Resisted user, taxi
Cancel a booking	Registered user

Taxi	
Responsibility	**Collaboration**
Book for a order	booking

Offer	
Responsibility	**Collaboration**
Adding offer	Booking

Payment	
Responsibility	**Collaboration**
Make new payment	Booking, Offer
Generate pay details	
Check payment details	
Confirm payment details	

Exercise 1


UnregisteredUser
#custId:int
#custName:char
#custaddress:char
#custemail:char
#custTel:int
+UnregisteredUser()
+UnregisteredUser(cid:int,cname:char,caddress:char,cmail:char,ctel:int)
+registration():void
+cancelregistration():void
+setcustId(cId:int):void
+setcustName(cname:char):void
+setaddress(caddress:char):void
+setemail(cmail:char):void
+displayUnreUser():void
~UnregisteredUser()
1
1..*
RegisteredUser
#userame:char
#password:char
+RegisteredUser()
+RegisteredUser(Uname:char,Upw:char)
+ login():void
+passwordvalidation():void
+setusername(Uname:char):void
+setpassword(Upw:char):void
+checkPassword():void
+displayRegUser():void
+logout():void
~RegisteredUser()
1
1..*
Booking
-BookingId;char
-StartPlace:char
-EndPlace:char
-date:char
-time:char
+Booking()
+Booking(Bid:char,Bstart:char,Bend:char,Bdate:char,Btime:char)
+addbookingdetails():void
+canclebookingdetails():void
+setbookingId(Bid:char):void
+setStartplace(Bstart:char):void
+setEndPlace(Bend:char):void
+setDate(Bdate:char):void
+setTime(Btime:char):void
+displaybooking():void
~Booking()
1..*
1
Taxi
-taxiNo:int
-taxiType:char
+Taxi()
+Taxi(Tno:int,Ttype:char)
+addtaxiInformation():void
+deletetaxiinformation():void
+setTaxiNo(Tno:int):void
+setTaxiType(Ttype:char):void
+displaytaxi():void
~Taxi()
1
0..*
Offer
-OfferId:char
-Offerdescription:char
+Offer()
+Offer(Oid:char,Odesc:char)
+addOffer():void
+deleteOffer():void
+updateOffer():void
+setofferId(Oid:char):void
+setofferdescription(Odesc:char):void
+displayOffer():void
~Offer()
0..*
1
Payment
-PayId:int
-PayType:char
-Amount:double
+Payment()
+Payment(Pid:int,Ptype:char,Pamount:double)
+setpayId(Pid:int):void
+setpayType(Ptype:char):void
+setamount(Pamount:double):void
+checkpaymenet():void
+confirmpaymet():void
+displaypayment():void
~payment()

Exercise 2

UnregisteredUser.h

```
class UnregisteredUser
{
protected:
      int custId;
      char custName[30];
      char custAddress[50];
      char custEmail[30];
      int custTelphone;
public:
      UnregisteredUser();
      UnregisteredUser(int cid, const char cname[], const char
caddress[], const char cmail[], int ctel);
      void setcustId();
      void setcustName();
      void setcustAddress();
      void setcustEmail();
      void setcustTelphone();
      void registration();
      void cancelregistration();
      void displayUnreUser();
      ~UnregisteredUser();
};
```

RegisteredUser.h

```
#include"UnregisteredUser.h"
#include"Booking.h"
class RegisteredUser : public UnregisteredUser
{
      protected:
            char username[10];
            char password[10];
      private:
            int NoofBooking;
            Booking* Book[2];
      public:
            RegisteredUser();
            RegisteredUser(const char Uname[], const char Upw[],int
cid, const char cname[], const char caddress[], const char cmail[],
int ctel);
            void setUsername();
            void setpassword();
            void login();
            void checkPassword();
            void passwordvalidation();
            void addBooking();
            void displayRegUser();
            void logout();
            ~RegisteredUser();
};
```

Booking.h

```
#include"RegisteredUser.h"
#include"Taxi.h"
#include"Offer.h"

class Booking
{
      private:
            char Bookingid[5];
            char StartPlace[20];
            char EndPlace[20];
            char Date[6];
            char time[4];

            //class relationship
            Taxi* tax[2];
            Offer* offers[2];
            int NoOfRegisteredUsers;
            RegisteredUser* Registered;

      public:
            Booking();
            Booking(const char Bid[], const char Bstart[], const char
Bend[], const char Bdate[], const char Btime[]);
            void setbookingid();
            void setStartplace();
            void setEndplace();
            void setDate();
            void setTime();
            void Addtaxi();
            void addbookingdetails();
            void cancelbookingdetails();
            void displaybookingdetails();
            ~Booking();
}
```

Taxi.h

```
class Taxi
{
      private:
            int taxiNo;
            char taxiType[10];
      public:
            Taxi();
            Taxi(int Tno, const char Ttype[]);
            void setTaxino();
            void setTaxitype();
            void addtaxiInformation();
            void deletetaxiInformation();
            void displaytaxi();
```

```
          ~Taxi();
};
```

Offer.h

```
#include"Payment.h"
class Offer
{
     private:
          char offerId[10];
          char offerdescription[30];

          Payment* pay[3];
     public:
          Offer();
          Offer(const char Oid[], const char Odesc[]);
          void setofferif();
          void setofferdescription();
          void addoffer();
          void addpayment();
          void updateoffer();
          void deleteoffer();
          void displayoffer();
          ~Offer();
};
```

Payment.h

```
class Payment
{
private:
     int Payid;
     char Paytype[30];
     double Amount;
public:
     Payment();
     Payment(int Pid, const char CPaytype[], double Pamount);
     void setpayid();
     void setPaytype();
     void setAmount();
     void checkpayment();
     void confirmpayment();
     void displaypayment();
     ~Payment();
};
```

UnregisteredUser.cpp

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

UnregisteredUser::UnregisteredUser()
{
      custId = 0;
      strcpy_s(custName, "");
      strcpy_s(custAddress, "");
      strcpy_s(custEmail, "");
      custTelphone = 0;
}
UnregisteredUser::UnregisteredUser(int cid, const char cname[], const
char caddress[], const char cmail[], int ctel)
{
      custId = cid;
      strcpy_s(custName, cname);
      strcpy_s(custAddress, caddress);
      strcpy_s(custEmail, cmail);
      custTelphone = ctel;
}

//setters
void UnregisteredUser::setcustId()
{
}

void UnregisteredUser::setcustName()
{
}

void UnregisteredUser::setcustAddress()
{
}

void UnregisteredUser::setcustEmail()
{
}

void UnregisteredUser::setcustTelphone()
{
}

//methods
void UnregisteredUser::registration()
{
}

void UnregisteredUser::cancelregistration()
{
}
```

```
void UnregisteredUser::displayUnreUser()
{
	cout << "this is unregisted user parent class" << endl;
}

//destructor
UnregisteredUser::~UnregisteredUser()
{
}
```

RegisteredUser.cpp

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

RegisteredUser::RegisteredUser()
{
	strcpy_s(username, "");
	strcpy_s(password, "");
}

RegisteredUser::RegisteredUser(const char Uname[], const char Upw[], 
int cid, const char cname[], const char caddress[], const char 
cmail[], int ctel)
{
	strcpy_s(username, Uname);
	strcpy_s(password, Upw);
	NoofBooking = 0;
}

//setters
void RegisteredUser::setUsername()
{
}

void RegisteredUser::setpassword()
{
}

//methods
void RegisteredUser::login()
{
}

void RegisteredUser::checkPassword()
{
}

void RegisteredUser::passwordvalidation()
{
}

void RegisteredUser::addBooking()
{
}

void RegisteredUser::displayRegUser()
{
}

void RegisteredUser::logout()
```

```
{
}

//destructor
RegisteredUser::~RegisteredUser()
{
}
```

Booking.cpp

```
#include "Booking.h"
#include"RegisteredUser.h"
#include"Taxi.h"
#include"Offer.h"
#include<iostream>
using namespace std;
#include<cstring>

Booking::Booking()
{
	strcpy_s(Bookingid, "");
	strcpy_s(StartPlace, "");
	strcpy_s(EndPlace, "");
	strcpy_s(Date, "");
	strcpy_s(time, "");
}

Booking::Booking(const char Bid[], const char Bstart[], const char
Bend[], const char Bdate[], const char Btime[])
{
	strcpy_s(Bookingid, Bid);
	strcpy_s(StartPlace, Bstart);
	strcpy_s(EndPlace, Bend);
	strcpy_s(Date, Bdate);
	strcpy_s(time, Btime);
	Registered->addBooking();
}

//setters
void Booking::setbookingid()
{
}

void Booking::setStartplace()
{
}

void Booking::setEndplace()
{
}

void Booking::setDate()
{
}

void Booking::setTime()
{
}

void Booking::Addtaxi()
{
}
```

```
void Booking::addbookingdetails()
{

}

void Booking::cancelbookingdetails()
{
}

void Booking::displaybookingdetails()
{
}

// destructor
Booking::~Booking()
{
}
```

Taxi.cpp

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

Taxi::Taxi()
{
	taxiNo = 0;
	strcpy_s(taxiType, "");

}

Taxi::Taxi(int Tno, const char Ttype[])
{
	taxiNo = Tno;
	strcpy_s(taxiType, Ttype);

}

//setters
void Taxi::setTaxino()
{
}

void Taxi::setTaxitype()
{
}

void Taxi::addtaxiInformation()
{
}

void Taxi::deletetaxiInformation()
{
}

void Taxi::displaytaxi()
{
}

//destructor
Taxi::~Taxi()
{
}
```

Offer.cpp

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

Offer::Offer()
{
	strcpy_s(offerId, "");
	strcpy_s(offerdescription, "");
}

Offer::Offer(const char Oid[], const char Odesc[])
{
	strcpy_s(offerId, Oid);
	strcpy_s(offerdescription, Odesc);
}

//setters
void Offer::setofferif()
{
}

void Offer::setofferdescription()
{
}

void Offer::addoffer()
{
}

void Offer::addpayment()
{
}

void Offer::updateoffer()
{
}

void Offer::deleteoffer()
{
}

void Offer::displayoffer()
{
}

//destructor
Offer::~Offer()
{
}
```

Payment.cpp

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

Payment::Payment()
{
	Payid = 0;
	strcpy_s(Paytype, "");
	Amount = 0.0;
}

Payment::Payment(int Pid, const char CPaytype[], double Pamount)
{
	Payid = Pid;
	strcpy_s(Paytype, CPaytype);
	Amount = Pamount;
}

//setters
void Payment::setpayid()
{
}

void Payment::setPaytype()
{
}

void Payment::setAmount()
{
}

void Payment::checkpayment()
{
}

void Payment::confirmpayment()
{
}

void Payment::displaypayment()
{
}

//destructor
Payment::~Payment()
{
}
```

Main.cpp

```
#include"UnregisteredUser.h"
#include"RegisteredUser.h"
#include"Booking.h"
#include"Taxi.h"
#include"Offer.h"
#include"Payment.h"
#include<iostream>
using namespace std;

int main()
{
	//object creation
	UnregisteredUser Ureg1;
	RegisteredUser reg1;
	Booking book1;
	Taxi* t1 = new Taxi;
	Offer* o1 = new Offer;
	Payment* p1 ;

	//methods implementation
	Ureg1.displayUnreUser();

	reg1.registration();
	reg1.login();

	book1.addbookingdetails();
	book1.cancelbookingdetails();

	t1->addtaxiInformation();
	t1->displaytaxi();

	o1->addoffer();
	o1->displayoffer();
	o1->addpayment();

	p1->checkpayment();
	p1->displaypayment();
};
```

Participation

Registration No	**Name**	**part**
IT21239434	P.H.K.C. Punchihewa	Registered User, Unregistered user
IT21258794	Karunasena H.G.M.K.K.L	Payment
IT21373770	Dissanayake D.M.U.I	Booking
IT21053696	Wickramasekara R.K.P	Offer
IT21209352	D.M.K.C. Bandara	Taxi