Sri Lanka Institute of Information Technology

Topic : car rental system

Group Number : MLB_9.1_g2

Campus : Malabe

Submission Date : 20/5/2022

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
IT21176906	N A M P DESHITHA	0768762050
IT21175220	D M P W KALUARACHCHI	0778331586
IT21176456	MALEESHA GIMMANA	0764146749
IT21174926	HASITHA JAGODA	0764106511
IT21178986	I A N M KARUNARATHNA	078488918

Object oriented concepts

B.Sc (Hons) in Information Technology

Exercise 1

System Requirements for Car Rental System

1. users can register as a Registered user .
2. registered user need to give his username , password to login to the system.
3. All the Registered users may freely edit their username and contact information. They are also able to delete their created account.
4. Registered users may view advanced details regarding vehicles and check their availability and condition.
5. Upon selecting a suitable vehicle, registered users can set the date and time they want to receive the vehicle and the duration it will be hired for along with the distance they estimate for their journey.
6. According to the availability determined using the information from clause 5 above, the system will display a list of drivers and driver's details for the registered user to select if necessary.
7. The registered user then confirms the reservation and is asked to confirm payment via using credit cards, PayPal or EFT.
8. The system offers another log in option which has admin privileges.
9. Admin also need to login to the system using his adminname and password.
10. The admin is given the authority to add and remove both vehicles and user account in the system. He may also view reservation details and update it's details.

SLIIT
Discover Your Future

BSc (Hons) in Information Technology

Assignment II

IT1050 – Object Oriented Concepts

Year 1, Semester II, 2022

Noun & Verb Analysis

- Nouns in RED color
- Verbs in BLUE color

1. users can register as a Registered user .

2. registered user need to give his username , password to login to the system.

3. All the Registered users may freely edit their username and contact information. They are also able to delete their created account.

4. Registered users may view advanced details regarding vehicles and check their availability and condition.

5. Upon selecting a suitable vehicle, registered users can set the date and time they want to receive the vehicle and the duration it will be hired for along with the distance they estimate for their journey.

6. According to the availability determined using the information from clause 5 above, the system will display a list of drivers and driver's details for the registered user to select if necessary.

7. The registered user then confirms the reservation and is asked to confirm payment via using credit cards, PayPal or EFT.
8. After the payment registered customer receive a bill.

9. The system offers another log in option which has admin privileges.

10. Admin also need to login to the system using his adminname and password.

11. The admin is given the authority to add and remove both vehicles and user account in the system. He may also view reservation details and update it's details and he can view payment details.

12. The admin also can remove the driver details in the system

13. Registered user can request to admin to remove the reservation.

Classes identified using Noun-Verb analysis

Identified Classes:

Registered User

Admin

Vehicle

Driver

Payment

Reservation

Nouns:

user - redundant for registered user

Comment- attribute of unregistered user

Username , password- attribute of registered user

Contact information- attribute of registered user

Account- outside the scope

Registered user-class

Vehicle-class

Availability-attribute of vehicle

Condition-attribute of vehicle

Date- attribute of reservation

Time-attribute of reservation

Duration- attribute of reservation

Distance-attribute of reservation

Driver-class

Bill-attribute of payement

Driver details-attribute of driver

Reservation-class

Payment method- attribute of payment

System- out of scope

Payment-class

Credit cars, paypal , EFT -attributes of payment

Admin-class

Adminname , password- attributes of admin

Reasons for rejecting other nouns

1. Redundant
    a. User is refers to the same person as 'Registered User'.
2. Outside scope of the system
    a. System is outside scope of the car rental system
    b. account
3. Attributes
    a. Comment
    b. Username , password
    c. Contact information
    d. Availability
    e. Condition
    f. Date
    g. Time
    h. Duration

i. Distance
j. Driver details
k. Payment method
l. Credit cars, paypal , EFT
m. Adminname , password
n. bill

Exercise 2

CRC Cards for Car rental system

Registered Customer	
Responsibilities:	Collaborations:
Login to the system	
Edit profile	
Delete profile	
View vehicle details	Vehicle
View availability of vehicles	Vehicle
View driver details	Driver
Select a vehicle	Reservation , vehicle
Select a driver	Driver
Request to delete the reservation	Admin
Receive a bill	payment
Pay for the reservation	reservation

Reservation	
Responsibilities:	Collaboration:

Update reservation	

Admin	
Responsibilities:	Collaboration:
Add or remove vehicles	Vehicle
Delete reservation	
Delete driver details	
View payment details	payment
View reservation details	reservation

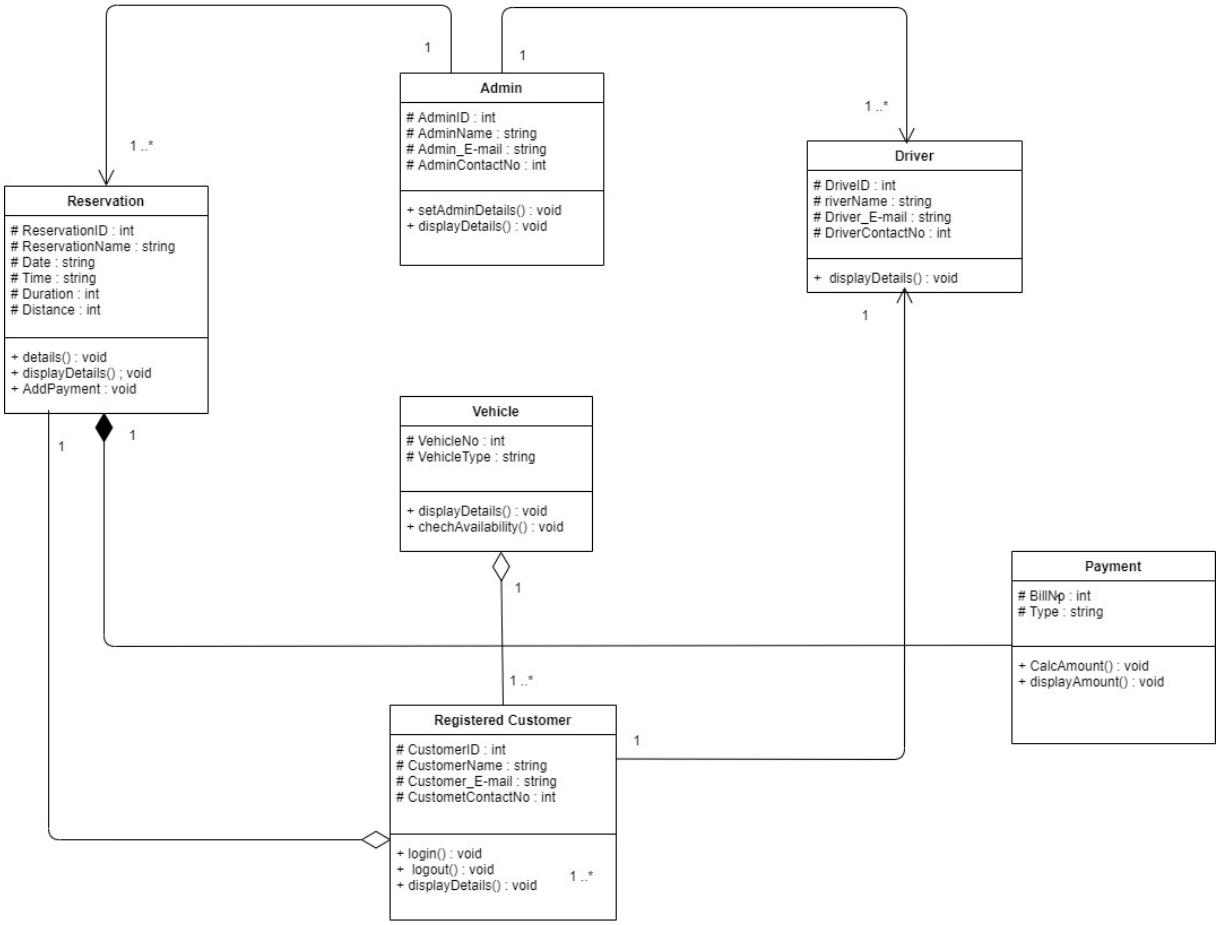
Driver	
Responsibilities:	Collaboration:
Store driver details	
Update driver	admin

Payment	
Responsibilities:	Collaboration:
Update payment details	Register user
Create bills	

Vehicle	
Responsibilities:	Collaboration:

Update the vehicle details	
Update vehicle availabilities	
Check reservation	Reservation

Class Diagram (UML Notation)



Class admin

```
class admin
{
private:
        int AdminID;
        string AdminName;
        string Admin_Email;
        int AdminContactNo;

public:
        void setAdminDetails(int id,string name,string mail,int number);
        void displayDetails();

};
```

class driver

```
class driver
{
private:
        int dID;
        string dName;
        string dEmail;
        int dContact;

public:
        void setDriverDetails(int id, string name, string email, int contact);
        void displayDriverDetails();
};
```

Class payment

```
class Payment //the class
{
private: //access
        int billNo;//variables
        string paymenttype;
        double amount;
public:
        Payment(int pbillNo, string ptype, double pamount);
        void calAmount();//code to be executed
        void displayPayment();//code to be executed
        ~Payment();//destructor
};
```

class registered Customer

```
class registeredCustomer //the class
{
private: //access
        reservation* reserv;
        string customerName;
        int customerID; //attribute
        string customerAddress;
        driver *d;

public:
        registeredCustomer(string name, int id, string address); //constructor with
parameters
        void addReservationDetails(reservation* reserv1);
        void displaycustomer(); //code to be executed
        ~registeredCustomer();
```

```
};
```

class reservation

```
class reservation//reservation class //whole class for payment class
{
private:
        string reserveName;
        int reserveId;
        string vehicleID;
        Payment* amount;

public:

        reservation(string name, int ID,string vehID);
        ~reservation();//destructor
```

```
        void displayReservation();
        void addPaymentdetails(int paybillNo,string paytype,double payamount);
        void displayPay();
};
```

class vehicle

```
class vehicle//class
{
  private:
        int vID;
        string vtype;


public:
        void setvehicleDetails(int id,string type);
  void vehicle::displayvehicleDetails();
        void checkAvailability();
};
```

Admin ccp

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

using namespace std;

void Admin::setAdminDetails(int id, string name, string mail, int number)
{
        AdminID = id;
        AdminName = name;
        Admin_Email = mail;
        AdminContactNo = number;
};
```

```
void Admin::displayDetails()
{
        cout << "Admin ID: " << AdminID << endl;
 cout << "Admin Name: " << AdminName << endl;
 cout << "E-mail: " << Admin_Email << endl;
 cout << "Contact Number: " << AdminContactNo << endl;
};
```

Driver ccp

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

using namespace std;

void driver::setDriverDetails(int id, string name, string email, int contact)
{
        dID = id;
        dName=name;
        dEmail=email;
        dContact = contact;
};
```

```
void driver::displayDriverDetails()
{

};
```

Payment ccp

```
#include <iostream>
#include <string>
#include "Payment.h"

Payment::Payment(int pbillNo, string ptype, double pamount)
{
        billNo = pbillNo;
        paymenttype = ptype;
```

```
		amount = pamount;
	}

	void Payment::calAmount()
	{

	}
	void Payment::displayPayment()
	{
		cout << "bill No: " << billNo << "\n" << "payment type: " << paymenttype << "\n" << "Amount: " << amount << endl;
	}
	Payment::~Payment()
	{
		cout << "delete billNo: " << billNo << endl;
	}
```

registeredCustomer ccp

```
#include <iostream>
#include <string>
#include "registeredCustomer.h"

using namespace std;
```

```
void registeredCustomer::addReservationDetails(reservation* reserv1)
{
        reserv = reserv1;
}

registeredCustomer::registeredCustomer(string name, int id, string address)
{
        customerName = name;
        customerID = id;
        customerAddress = address;
}

void registeredCustomer::displaycustomer()
{
        cout << "cutomer Name: " << customerName << endl;
        cout << "cutomer ID: " << customerID << endl;
        cout << "customer Address: " << customerAddress << endl;

        reserv->displayReservation();

}

registeredCustomer::~registeredCustomer()
{
        cout << "delete registered customer" << endl;
}
```

SLIIT
Discover Your Future

BSc (Hons) in Information Technology

Assignment II

IT1050 – Object Oriented Concepts **Year 1, Semester II, 2022**

Reservation ccp

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

```
using namespace std;


reservation::reservation(string name, int ID,string vehID)
{
	reserveName = name;
	reserveId = ID;
	vehicleID = vehID;
}


reservation::~reservation()
{
	cout << "delete reservation" << endl;
	delete amount;
}


void reservation::displayReservation()
{
	cout << "reservation name: " << reserveName << endl;
	cout << "reservation ID: " << reserveId << endl;
}
void reservation::addPaymentdetails(int paybillNo, string paytype, double payamount)
{
	amount =new Payment(paybillNo,paytype,payamount);
}
void reservation::displayPay()
{
	amount->displayPayment();
}
```

SLIIT
Discover Your Future

BSc (Hons) in Information Technology

Assignment II

IT1050 – Object Oriented Concepts **Year 1, Semester II, 2022**

Vehicle ccp

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

using namespace std;

void vehicle::setvehicleDetails(int id, string type)
{
        vID = id;
        vtype=type;

};
void checkAvailability(){

};
void vehicle::displayvehicleDetails()
{

};
```

Main ccp

```
#include "Admin.h"

#include "Driver.h"

#include "Payment.h"

#include "reservation.h"

#include "vehicle.h"

#include "registeredCustomer.h"

#include <iostream>

using namespace std;

int main()
```

```
{
//---- Object creation
Admin admin1;
Admin* adminptr = &admin1;
Driver driver1;
Driver* driverptr = &driver1;
Payment payment1;
Payment* paymentptr=&payment1;
reservation res1;
reservation* reservtionptr=&res1;
vehicle veh1;
vehicle* vehicleptr=&veh1;
registeredCustomer reg1;
registeredCustomer* registeredCustomerptr=&reg1;



//----Method Calling------
adminptr->displayDetails();


Driverptr->displayDriverDetails();


Paymentptr->calAmount();
Paymentptr->displayPayment();


reservationptr->displayPay();
reservationptr->displayReservation();


vehicleptr->displayvehicleDetails();
```

```
registeredCustomerptr->~registeredCustomer();
registeredCustomerptr->displaycustomer();



 //----Delete Dynamic objects------
 delete admin1;
 delete driver1;
 delete payment1;
 delete res1;
 delete veh1;
 delete reg1;

 return 0;
}
```