Topic: Car Rental System

Group no : MLB_01.01_11

Campus: Malabe

Submission Date: 16/05/2022

SLIIT
Discover Your Future

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 Number	**Student Name**	**Contact Number**
1	IT21142628	D.M. Guwani Diwya Dissanayake	0741503175
2	IT21126574	Senadheera Dummunnage Lakshitha Sudath Perera	0767984052
3	IT21112850	K.G. Yoman Bisanda	0770162945
4	IT21114830	Thedas Sri Harisha P.A. D	0788186673
5	IT21321672	Ratwatte C.S. S	0769222562

Description

The Car Rental System is based on the idea of renting cars and having a firm generating rental invoices.

A user must first log in to gain access to the main system; after that, only the user can select automobiles of various models and rent them for specific days.

In terms of the car rental system's features, after logging in as a user, the user must provide a name before selecting available cars. After picking a vehicle, the system displays selected vehicle information, such as maximum power, mileage, and other factors. Next, the user must provide information such as the car number and the number of days for which the car will be rented. Following these steps, the system calculates rent and presents the Customer Invoice, which includes the invoice number, the customer's name, the car model, the number of days, and the total rental cost.

This mini-project has only a few features, but they are all necessary.

Tasks

Requirements

1. Firstly, the system should register the vehicles namely, cars, trucks, SUVs, vans, and motorcycles.

2. Every vehicle must be added with the registration number, model, color, manufacture year, and mileage. In addition, the system will give a vehicle id for each registered vehicle.

3. As a new Customer, he needs to first register. Provide registration details such as name, address, and mail.
4. Customer search for a vehicle and the system displays the car details. In addition, it displays the availability.
5. Customers can order a car through the car rental system.
6. The system will provide a reservation id and the details about the pick-up location, return location, as well as when the vehicle is nearing the due date and the return date.
7. Customers can inform the driver of the pick-up location and drop location they want to go to.
8. Customer can choose vehicle type van, car, and bus as their wants.
9. Customers should pay for their reservation using available payment.
10. After payment, the Customer receives the payment and sends messages.
11. After reserving an order. The driver confirms the ride.
12. Reservation can be not allowed as the requirement of the customer.
13. All the details of the system can manage by the admin.

The identified classes

1. User class

2. Driver class

3. Customer class

4. car rental location class

5. Vehicle class

6. Payment class

7. Reservation class

8. Admin class

9. Time Schedule class

10. car insurance class

14. CRC cards

1. User class

Class Name: -user	
Responsibilities	**Collaborators**
Register to the system	

2. Driver class

Class Name: -driver	
Responsibilities	**Collaborators**
A car schedule	vehicle
Add/ remove drivers	booking

3. Vehicle class

Class Name: - vehicle	
Responsibilities	**Collaborators**
Register to vehicle details the system	vehicle
Receive a vehicle id	
Display vehicle details.	
Displays the vehicle availability.	

4. Vehicle Reservation class

Class Name: - vehicle	
Responsibilities	**Collaborators**
Display the reservation id and the other details.	

5. Customer class

Class Name: - order	
Responsibilities	**Collaborators**
Register to the system	
Inform the driver	Pick-up location
Choose vehicle	
Pay	
Cancel reservation	reservation

6. Car rentaltal class

Class Name: - car rental class	
Responsibilities	**Collaborators**
Registration details	
Schedule time	

7. Payment class

Class Name: -payment	
Responsibilities	**Collaborators**
Customers select a payment method	

8. admin class

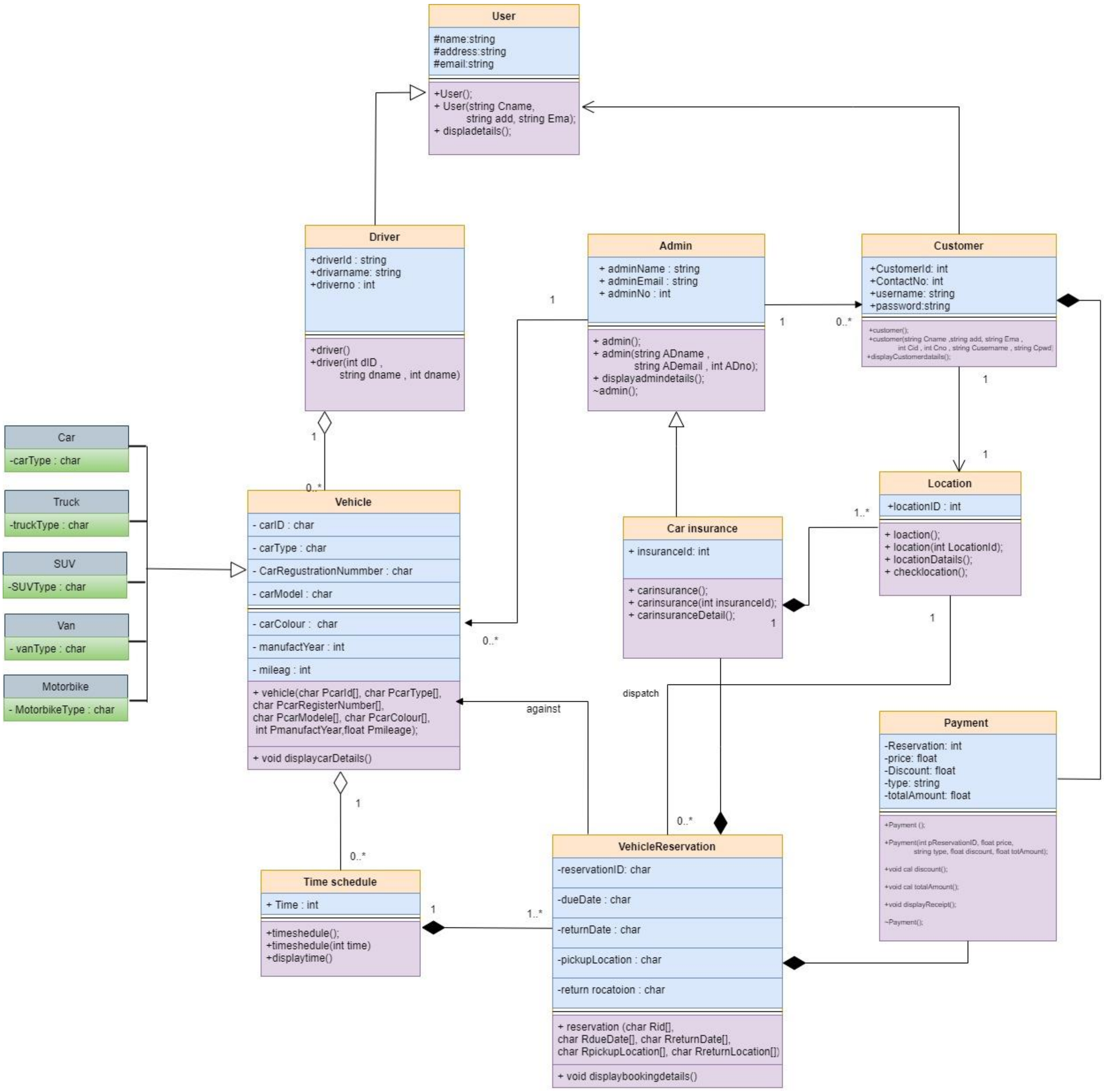
Class Name: - admin	
Responsibilities	**Collaborators**
System can manage	Driver
Add/ remove the driver	Vehicles
	Booking

9. time schedule class

Class Name: - time schedule	
Responsibilities	**Collaborators**
Store time and location details store	

10. car Insurance class

Class Name: - classInsurance class	
Responsibilities	**Collaborators**


User
#name:string
#address:string
#email:string
+User();
+ User(string Cname,
string add, string Ema);
+ displadetails();
Driver
+driverId : string
+drivarname: string
+driverno : int
+driver()
+driver(int dID ,
string dname , int dname)
Admin
+ adminName : string
+ adminEmail : string
+ adminNo : int
+ admin();
+ admin(string ADname ,
string ADemail , int ADno);
+ displayadmindetails();
~admin();
Customer
+CustomerId: int
+ContactNo: int
+username: string
+password:string
+customer();
+customer(string Cname ,string add, string Ema ,
int Cid , int Cno , string Cusername , string Cpwd)
+displayCustomerdatails();
1
1
0..*
1
1
Car
-carType : char
Truck
-truckType : char
SUV
-SUVType : char
Van
- vanType : char
Motorbike
- MotorbikeType : char
1
0..*
Vehicle
- carID : char
- carType : char
- CarRegustrationNummber : char
- carModel : char
- carColour : char
- manufactYear : int
- mileag : int
+ vehicle(char PcarId[], char PcarType[],
char PcarRegisterNumber[],
char PcarModele[], char PcarColour[],
int PmanufactYear,float Pmileage);
+ void displaycarDetails()
0..*
Car insurance
+ insuranceId: int
+ carinsurance();
+ carinsurance(int insuranceId);
+ carinsuranceDetail();
1
Location
+locationID : int
+ loaction();
+ location(int LocationId);
+ locationDatails();
+ checklocation();
1..*
1
against
dispatch
Payment
-Reservation: int
-price: float
-Discount: float
-type: string
-totalAmount: float
+Payment ();
+Payment(int pReservationID, float price,
string type, float discount, float totAmount);
+void cal discount();
+void cal totalAmount();
+void displayReceipt();
~Payment();
1
0..*
Time schedule
+ Time : int
+timeshedule();
+timeshedule(int time)
+displaytime()
1
1..*
0..*
VehicleReservation
-reservationID: char
-dueDate : char
-returnDate : char
-pickupLocation : char
-return rocatoion : char
+ reservation (char Rid[],
char RdueDate[], char RreturnDate[],
char RpickupLocation[], char RreturnLocation[])
+ void displaybookingdetails()

coding

// OOC assigment 2

// topic: MLB_01.01_11 car rental system

// D.M. Guwani Diwya DissanayakeD.M - IT21142628

```
#include<iostream>

#include<string>

using namespace std;
```

Customer

```
class Customer

{

        private:

                Payment *pay;

                Reservation * reservation;


                int CustomerId;

                int ContactNO;

                string userName;

                string password;

                string CustomerAdd;

        public:

                Customer() {};


                Customer(int pReservationID, double pprice, string type, float totAmount)

                {

                        pay = new Payment(pReservationID, pprice, type, totAmount);

                }
                void displayDisplayPayment()

                {

                        pay -> displayReceipt();

                }



                Customer(int pReservationID, const char pReservationType[], const char
pReservationName[], double price)

                {

                        reservation = new Reservation(pReservationID, pReservationType,
pReservationName, price);

                }



                void dispalayDisplayReservation()

                {

                        reservation -> displayReservation();

                }



                Customer(int cId, int cNo, string uname, string Cpw, string cAdd)

                {

                        CustomerId = cId;

                        ContactNO = cNo;

                       userName = uname;

                       password = Cpw;

                       CustomerAdd = cAdd;



                   }



                   void displayCustomer()

                   {
                        }

                         ~Customer()

   {

     delete pay;

     delete reservation

   }

};
```

Payment

```
#include<iostream>

#include<string>

using namespace std;


class Payment

{

            private:

                        int ReservationID;

   float price;

   string type;

   float Discount;

   float totalAmount;


            public:

   Payment();


   Payment(int pReservationID, float pprice, string type,

   float discount, float totAmount){
     ReservationID = pReservationID;
     price = pprice;
     type = type;
     Discount = discount;
     totalAmount = totAmount;
   }

   void calcDiscount()
   {
     Discount= (price * 0.1);
     cout<<"Discount amount is: "<< Discount << endl;
   }

   float calctotalAmount(){
     return (price- Discount);
   }
           void calctotalAmount(){
     totalAmount= price * 0.1;
     cout<< "Total Price is: "<< totalAmount << endl;
   }

   void displayReceipt()
   {


   }
   ~Payment(){};
};
```

//OOC Assingment 2
//Topic : MLB_01.01_11 Car Rental system
//Senadheera Dummunnage Lakshitha Sudath Perera   IT21126574

```
// User, customer, location ,car insurance

// use of overloaded Constructors
// The default constructor has sample values set once

#include<iostream>
#define SIZE 2
using namespace std;


class customer;

class admin;

class location;

class carinsurance;
```

User.h

```
class User{ //User class

   protected :
                  string name;
                  string address;
                  string email;

          public :

             User();//default constructor
             User(string Cname,string add, string Ema);//overloaded constructor
             void displadetails();
};
```

User.cpp

```
User::User()
{
        cout<< endl << "User Class" << endl;
}

User::User(string Cname ,string add, string Ema){

        name=Cname;
        address=add;
        email=Ema;
}

void User::displadetails(){

}
```

Customer.h

```
class  customer: public User{  // admin class

        private :
     int CustomerId;
     int ContactNo;
     string username;
     string password;
     location* location;
  private :

                 customer();  //default constructor
                 customer(string Cname ,string add, string Ema ,int Cid , int Cno , string
Cusername , string Cpwd);  // overloaded constructor
                 void displayCustomerdatails();

};
```

Customer.cpp

```
customer::customer(){

        cout << "Customer Class" << endl;
}
customer::customer(string Cname ,string add, string Ema ,int Cid , int Cno , string
Cusername , string Cpwd):User(Cname ,add,  Ema)
{
        name=Cname;
        address=add;
        email=Ema;
        CustomerId=Cid;
        ContactNo=Cno;
        username=Cusername;
        password=Cpwd;
}
void customer::displayCustomerdatails(){

        cout<< "<!------------------ Customer Details ---------------------------!>";

        cout<< "Name    :" << name<<endl;
        cout<< "Address :" << address<<endl;
        cout<< "Email   :" << email <<endl;
        cout<< "Customer ID : " << CustomerId <<endl;
        cout<< "Conatact Number :"<< ContactNo <<endl;
        cout<< "User Name :" <<username <<endl;
        cout<< "Password  : "<<password <<endl;



        cout<<
"<*************************************************************>";

}
```

Admin.h

```
class admin  { // admin class

        protected :

            string adminName;
            string adminEmail;
            int adminNo;
            customer* customers;//an object of Customer as attribute
        public :

                admin(); //default constructor
                admin(string ADname ,string ADemail , int ADno);// overloaded constructor
                void displayadmindetails();
                ~admin();

};
```

Admin.cpp

```
admin::admin(){

        cout<< endl << "Admin class" << endl;
}

admin::admin(string ADname ,string ADemail , int ADno)
{

}
void admin::displayadmindetails(){

        cout<< "<!------------------ Admin Details -------------------------------!>";

        cout<< "Admin Name   :"<<adminName <<endl;
        cout<< "Admin Email  :"<<adminEmail<<endl;
        cout<< "Admin Contact Number :"<<adminNo <<endl;

        cout<<
"<********************************************************************>";
}
admin::~admin(){

        cout << "Manage Details" << endl;
}
```

Location.h

```
class location: public admin{ // class location

        private :
                int locationId;

        public  :
                location();//default constructor
                location(int LocationId);// overloaded constructor
                void locationDatails();
                void checklocation();
};
```

Location.cpp

```
location::location(){

        cout<< "location Class" <<endl;
}
location::location(int LocationId){

  locationId=LocationId;

}
void location::locationDatails()
{
        cout<< "<!------------------------ location --------------------------------!>";


        cout<< "location ID : "<< locationId <<endl;


  cout<<
"<**********************************************************************>";

}
```

Carinsurance.h

```
class carinsurance { // class carinsurance

        private:
                int insuranceId;
                Reservation * Reservation[SIZE];
        public:

                carinsurance();//default constructor
                   carinsurance(int InsuranceId);//overloaded Constructors
                   void carinsuranceDetail();

};
```

Carinsurance.cpp

```
carinsurance::carinsurance(){
        carinsurance[0] = new carinsurance( );
          carinsurance[1] = new carinsurance( );
}
carinsurance::carinsurance(int InsuranceId){

        insuranceId=InsuranceId;
}
void carinsurance::carinsuranceDetail();
}
```

// OOC Assingment 2
// Topic: MLB_01.01_11 Car Rental System (vehicle & vehicle reservation)
// Ratwatte C.S. S   IT21321672

Vehicle Class

car.h

```
class vehicle
{
  private:
    char carId[10];
    char carType[20];
    char carRegisterNumber[10];
    char carModele[50];
    char carColour[20];
    int manufactYear;
    float mileage;

  public:
    vehicle(char PcarId[], char PcarType[], char PcarRegisterNumber[], char PcarModele[],
char PcarColour[], int PmanufactYear,float Pmileage);
    void displaycarDetails();

};
```

car.cpp

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

using namespace std;

vehicle :: vehicle(char PcarId[],  char PcarType[],char PcarRegisterNumber[],
char PcarModele[], char PcarColour[], int PmanufactYear, float Pmileage)
{
  strcpy(carId,PcarId);
  strcpy(carType,PcarType);
  strcpy(carRegisterNumber,PcarRegisterNumber);
  strcpy(carModele,PcarModele);
  strcpy(carColour,PcarColour);
  manufactYear=PmanufactYear;
  mileage=Pmileage;
}

void vehicle :: displaycarDetails()
{
  cout<<"Vehicle Id: "<<carId<<endl;
  cout<<"Vehicle Type: "<<carType<<endl;
  cout<<"Vehicle Register Number: "<<carRegisterNumber<<endl;
  cout<<"Vehicle Model: "<<carModele<<endl;
  cout<<"Vehicle Colour: "<<carColour<<endl;
  cout<<"Vehicle Manufacture Year : "<<manufactYear<<endl;
  cout<<"Vehicle Mileage : "<<mileage<<"km"<<endl;

}
```

Vehicle Reservation class

vreservation.h

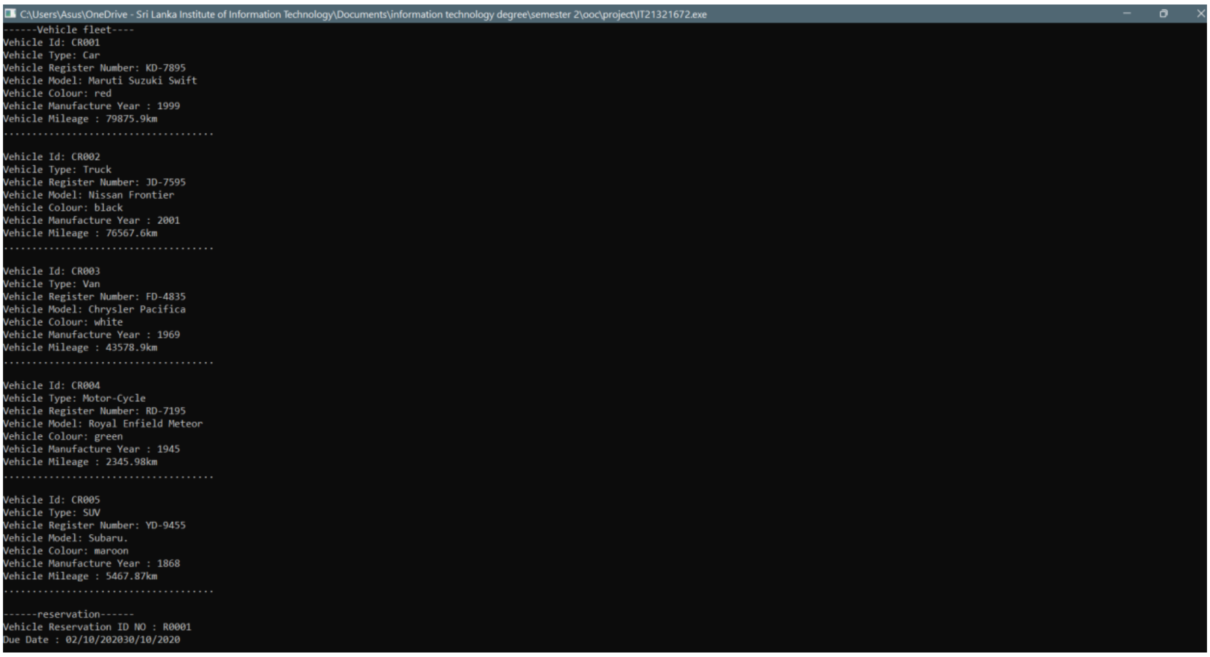
```
class reservation
{
  private:
    char reservationID[10];
    char dueDate[10];
    char returnDate[11];
    char pickupLocation[20];
    char returnLocation[20];

  public:
    reservation (char Rid[], char RdueDate[], char RreturnDate[],char RpickupLocation[], char
RreturnLocation[]);
    void displaybookingdetails();
};
```

vreservation.cpp

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

reservation::reservation (char Rid[], char RdueDate[], char RreturnDate[], char
RpickupLocation[], char RreturnLocation[])
{
  strcpy(reservationID,Rid);
  strcpy(dueDate,RdueDate);
  strcpy(returnDate,RreturnDate);
  strcpy(pickupLocation,RpickupLocation);
  strcpy(returnLocation,RreturnLocation);
}

void reservation::displaybookingdetails()
{
    cout << "Vehicle Reservation ID NO : "<< reservationID<< endl;
    cout << "Due Date : " << dueDate << endl;
    cout << "Return Date :" << returnDate << endl;
    cout << "Pickup Location :" << pickupLocation <<endl;
    cout << "Return Location :" << returnLocation <<endl;
}
```

main program

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

int main (){

    cout<<"------Vehicle fleet----"<<endl; //deatils about vehicles

    vehicle *car;
  car= new vehicle((char*)"CR001", (char*)"Car",(char*)"KD-
7895",  (char*)"Maruti Suzuki Swift", (char*)"red",1999,79875.9);
  car->displaycarDetails();
  cout<<"....................................."<<endl;
  cout<<endl;

  vehicle *truck;
  truck= new vehicle((char*)"CR002", (char*)"Truck",(char*)"JD-
7595",  (char*)"Nissan Frontier", (char*)"black",2001,76567.56);
  truck->displaycarDetails();
  cout<<"....................................."<<endl;
  cout<<endl;

  vehicle *van;
  van= new vehicle((char*)"CR003", (char*)"Van",(char*)"FD-
4835",  (char*)"Chrysler Pacifica", (char*)"white",1969,43578.87);
  van->displaycarDetails();
  cout<<"....................................."<<endl;
  cout<<endl;

  vehicle *motorcycle;
  motorcycle= new vehicle((char*)"CR004", (char*)"Motor-Cycle",(char*)"RD-
7195",  (char*)"Royal Enfield Meteor ", (char*)"green",1945,2345.98);
  motorcycle->displaycarDetails();
  cout<<".....................................";<<endl;
  cout<<endl;

  vehicle *SUV;
  SUV= new vehicle((char*)"CR005", (char*)"SUV",(char*)"YD-
9455",  (char*)"Subaru.", (char*)"maroon",1868,5467.87);
  SUV->displaycarDetails();
  cout<<".....................................";<<endl;
  cout<<endl;



 cout<<"------reservation------"<<endl; // details about vehicle reservations

    reservation *book1;
    book1 = new reservation((char*)"R0001", (char*)"02/10/2020", (char*)"30/10
/2020",(char*)"Matale", (char*)"Kiribathkubura");
    book1 -> displaybookingdetails();
    cout<<".....................................";<<endl;
    cout<<endl;

    reservation *book2;
    book2 = new reservation((char*)"R0002", (char*)"02/12/2020", (char*)"02/10
/2020",(char*)"Matara", (char*)"Galle");
    book2 -> displaybookingdetails();
    cout<<".....................................";<<endl;
    cout<<endl;

    reservation *book3;
    book3 = new reservation((char*)"R0003", (char*)"02/12/2021", (char*)"02/10
/2021",(char*)"Kandy", (char*)"Kurunagala");
    book3 -> displaybookingdetails();
    cout<<".....................................";<<endl;
    cout<<endl;

    reservation *book4;
    book4 = new reservation((char*)"R0004", (char*)"02/11/2021", (char*)"03/10
/2021",(char*)"Anuradapura", (char*)"Kurunagala");
    book4 -> displaybookingdetails();
    cout<<".....................................";<<endl;
    cout<<endl;
}
```



C:\Users\Asus\OneDrive - Sri Lanka Institute of Information Technology\Documents\information technology degree\semester 2\ooc\project\IT21321672.exe

```
------Vehicle fleet----
Vehicle Id: CR001
Vehicle Type: Car
Vehicle Register Number: KD-7895
Vehicle Model: Maruti Suzuki Swift
Vehicle Colour: red
Vehicle Manufacture Year : 1999
Vehicle Mileage : 79875.9km
.....................................

Vehicle Id: CR002
Vehicle Type: Truck
Vehicle Register Number: JD-7595
Vehicle Model: Nissan Frontier
Vehicle Colour: black
Vehicle Manufacture Year : 2001
Vehicle Mileage : 76567.6km
.....................................

Vehicle Id: CR003
Vehicle Type: Van
Vehicle Register Number: FD-4835
Vehicle Model: Chrysler Pacifica
Vehicle Colour: white
Vehicle Manufacture Year : 1969
Vehicle Mileage : 43578.9km
.....................................

Vehicle Id: CR004
Vehicle Type: Motor-Cycle
Vehicle Register Number: RD-7195
Vehicle Model: Royal Enfield Meteor
Vehicle Colour: green
Vehicle Manufacture Year : 1945
Vehicle Mileage : 2345.98km
.....................................

Vehicle Id: CR005
Vehicle Type: SUV
Vehicle Register Number: YD-9455
Vehicle Model: Subaru.
Vehicle Colour: maroon
Vehicle Manufacture Year : 1868
Vehicle Mileage : 5467.87km
.....................................

------reservation------
Vehicle Reservation ID NO : R0001
Due Date : 02/10/202030/10/2020
```

// OOC assigment 2
// topic : MLB_01.01_11 car rental system
// Thedas Sri Harisha P.A. D   IT21114830

```
#include<iostream>
#include<cstring>
using namespace std;
```

```
class driver;
class timeshedule;
```

driver.h

```
class driver{


        private:
                int driverid;
                string drivername;
                int driverno;
        public :
          driver();
    driver(int dID ,string dname,int dno);
    void diplaydriverdetals();

};
```

driver.cpp

```
driver::driver()
{
        cout << "class Driver" << endl;
}

driver::driver(int dID ,string dname , int dno)
{
        driverid=dID;
        drivername=dname;
        driverno=dno;
}
```

```
void driver::diplaydriverdetals(){

        cout<< "Driver ID : "<< driverid <<endl;
        cout<< "Driver Name : " <<drivername<<endl;
        cout<< "Driver Contact Number :" << driverno <<endl;
}
```

timeshedule.h

```
class timeshedule{

        private :
                int time;

  public :

     timeshedule();
     timeshedule(int time);
     void displaytime();
};
```

timeshedule.cpp

```
timeshedule::timeshedule(){

        cout << "class Time shedule"<<endl;
}
timeshedule::timeshedule(int ptime){

        time=ptime;
}
void timeshedule::displaytime(){

        cout<<" Pick-Time :" <<time<<endl;
```

Individual contribution.

	Registration Number	Exercise 01	Exercise 02
1	IT21142628	Payment Customer	Payment Customer
2	IT21126574	Customer Location Insurance User admin	Customer Location Insurance User admin
3	IT21112850	-	-
4	IT21114830	Time schedule driver	Time schedule driver
5	IT21321672	Vehicle Vehicle reservation	Vehicle Vehicle reservation