



**Topic : Vehicle Rental System**

**Group no : MLB\_02.02\_05**

**Campus : Malabe**

**Submission Date : 2022/05/18**

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
IT21214820	Dharmathilake K.A.D.K.D.	0702534675
IT21214516	Fernando N.K.B	0715689128
IT21213144	Senarathne D.R.	0770119377
IT21215810	Pathirana G.P.M.P.	0719659889
IT21216596	Thihansa K.A.S.	0714621934

## Requirements of the System

- 1. User registration:** Before using the system, the user must first register. According to the function of each user, the system will have many types of users. The user needs to input their name, contact number, email address, and permanent address for registration.
- 2. User Login:** After registering, the user must log in to use other features such as booking and providing feedback.
- 3. Search vehicles:** Users can search vehicles and select the preferred vehicle type to reserve a booking.
- 4. Reservation of vehicle:** After selecting a vehicle, the user can reserve it. However, before beginning the reservation procedure, the user must first register. The registered user must fill out reservation information such as the pickup date, time, rental duration, and location and send an inquiry.
- 5. Check the availability of vehicle:** of the vehicle: After the user sends the inquiry employee of the rental company staff will check the availability of that specific vehicle. If the vehicle is not available, the staff will send a message to do another reservation to the user.
- 6. Request driver:** The registered driver can request a driver from the company alongside the reservation
- 7. Calculate payment:** Once the reservation is done, the accountant will calculate the payment after considering the reservation duration and vehicle rental charge.
- 8. Make payment:** After the reservation, the registered user can do the payment.
- 9. Cancel reservation:** If the user is not satisfied with the reservation, the registered user can cancel the reservation within 7 days.
- 10. Add discount:** Considering the reservation history of the user, an accountant can add a discount to the payment.
- 11. Send feedback:** The user can send feedback after receiving the service of the rental company. They could also update the feedback at any time.
- 12. View feedback:** Admin can view feedbacks that was sent by the registered user.
- 13. Reply to feedback:** Admin can send a reply to the feedback if necessary.
- 14. Add a new vehicle:** Admin can add new vehicles to the system. Admin needs vehicle details for this process.
- 15. Update vehicle details:** Admin can update details regarding vehicles.
- 16. Assign driver:** If a registered user requests a driver, the admin must assign one.

**17. Calculate the salary of the employees:** The accountant must calculate the salary at the end of each month.

**18. Generate report:** At the end of the month, a report will be generated including the reservation details for the month

**19. View report:** Once the report is generated staff of the rental company can view it.

**20. Update user details:** Registered users can update user details at anytime

**21. Check payment:** Employees must verify the payment of the registered user.

## **Classes**

- 1.** User
- 2.** Employee
- 3.** Accountant
- 4.** Admin
- 5.** Driver
- 6.** Reservation
- 7.** Vehicle
- 8.** Payment
- 9.** Discount
- 10.** Feedback
- 11.** Report

# CRC Cards

User

User Class	
Responsibilities	Collaborations
Update user details	
Register user	
User registration	

Vehicle

Vehicle Class	
Responsibilities	Collaborations
Check availability	Reservation
Add vehicle details	
Update vehicle details	
Search vehicles	

Reservation

Reservation Class	
Responsibilities	Collaborations
Make a reservation	
Cancel reservation	

Driver

Driver Class	
Responsibilities	Collaborations
Add driver details	
Update driver details	
Request driver	

## Employee

Employee Class	
Responsibilities	Collaborations
Check payment	Payment, Reservation

## Accountant

Accountant Class	
Responsibilities	Collaborations
Calculate payment	Payment
Calculate employee salary	

## Feedback

Feedback Class	
Responsibilities	Collaborations
Send feedback	
Remove feedback	
Update feedback	

## Payment

Payment Class	
Responsibilities	Collaborations
Make payment	
Pay payment	

## Admin

Admin Class	
Responsibilities	Collaborations
Assign driver	Driver
Reply to feedback	Feedback
View feedback	Feedback

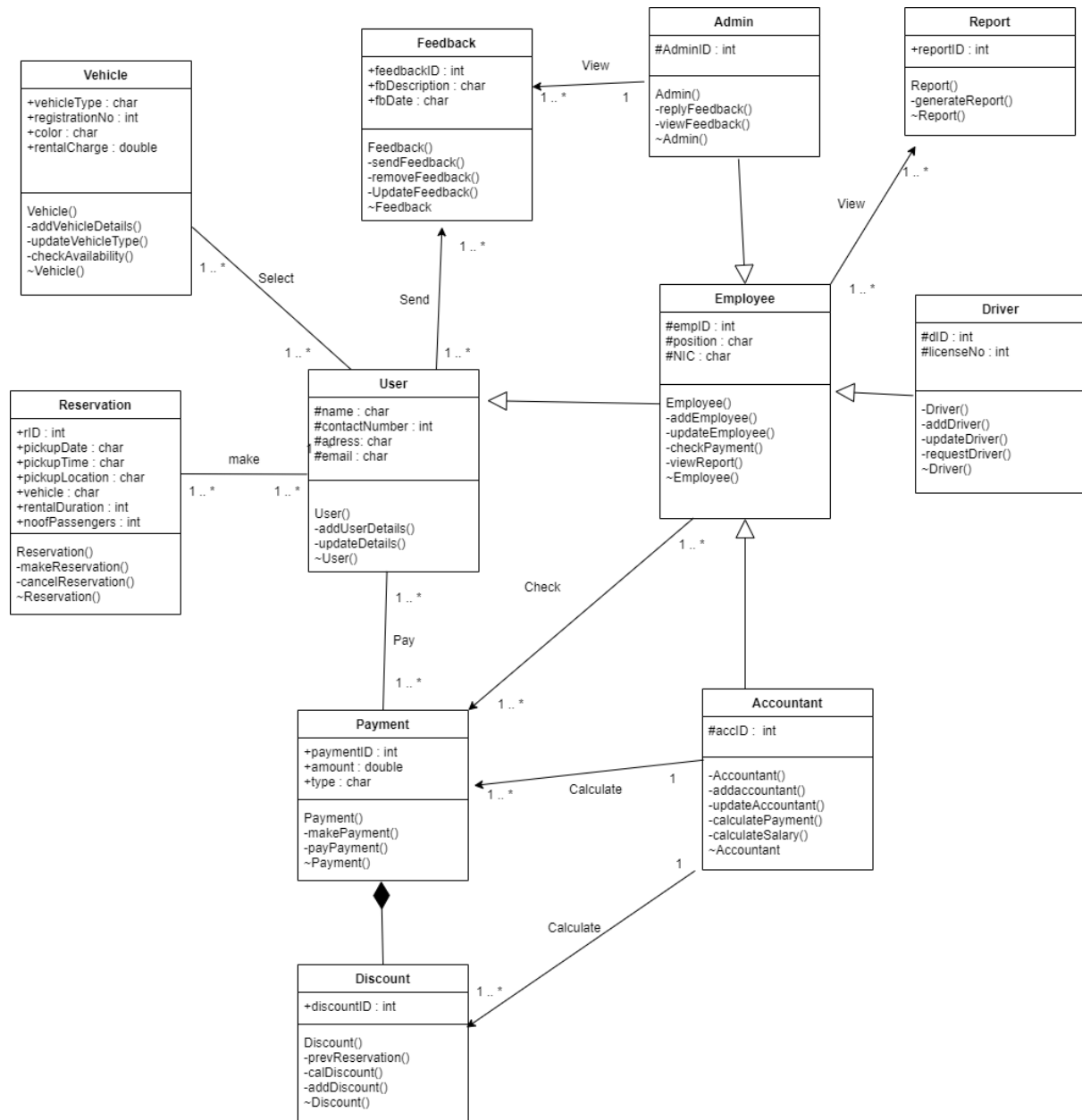
Report

Report Class	
Responsibilities	Collaborations
Generate report	Reservation

Discount

Discount Class	
Responsibilities	Collaborations
Calculate discount	
Add discount to payment	Payment

# Class Diagram



## Coding

//MLB\_02.02\_05\_VehicleRentalSystem

//IT21214820

//User class

#include <iostream>

#include <cstring>

#include "Feedback.h"

#include "Reservation.h"

#include "Payment.h"

#include "Vehicle.h"

using namespace std;

class User

{

protected:

char name[30];

int contactNumber;

char email[50];

char adress[100];

Reservation\* res[5];

Payment\* pay[5];

Vehicle\* vehicle[5];

Feedback\* fb[5];

public:

User()

{



```
strcpy(name, "");  
contactNumber = 0;  
strcpy(email, "");  
strcpy(adress, "");  
};
```

```
void addUserDetails(char uname[30], int ucontactNumber, char uemail[50],char adress[100]);
```

```
void updateDetails(char uname[30], int ucontactNumber, char uemail[50],char adress[100]);
```

```
~User()  
{  
    cout << "User " << name<< "deleted" <<endl;  
};
```

```
};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21213144
```

```
//Employee class
```

```
#include <iostream>
```

```
#include <cstring>
```

```
#include "Report.h"
```

```
#include "Payment.h"
```

```
#include "User.h"
```

```
using namespace std;
```

```
class Employee: public User
```

```
{
```

```
protected:
```

```
    int empID;
```

```
    char position[15];
```

```
    char NIC[12];
```

```
    Report* rep[5];
```

```
    Payment* pay[5];
```

```
public:
```

```
    Employee()
```

```
{
```

```
    strcpy(name, "");
```

```
    contactNumber = 0;
```

```
    strcpy(email, "");
```

```
    strcpy(adress, "");
```

```
    empID = 0;
```

```

        strcpy(position, "");

        strcpy(NIC, "");

    };

    void addEmployee(const char ename[30], int econtactNumber, const char Eemail[50],const char
eadress[100], int employeeID, const char ePosition[15], const char nicNo[12]);

    void updateEmployee(int employeeID, const char ePosition[15], const char nicNo[12]);

    void checkPayment();

    void viewReport();

    ~Employee()
    {
        cout << "Employee " << empID << " deleted " <<endl;
    };
};

```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21213144
```

```
//Accountant class
```

```
#include <iostream>
```

```
#include <cstring>
```

```
#include "Payment.h"
```

```
#include "Discount.h"
```

```
#include "Employee.h"
```

```
using namespace std;
```

```
class Accountant : public Employee
```

```
{
```

```
protected:
```

```
    int accID;
```

```
    Payment* pay[5];
```

```
    Discount* dis[5];
```

```
public:
```

```
    Accountant()
```

```
{
```

```
    strcpy(name, "");
```

```
    contactNumber = 0;
```

```
    strcpy(email, "");
```

```

        strcpy(adress, "");

        strcpy(position, "");

        strcpy(NIC, "");

        accID = 0;

    };

    void addAccountant(const char aname[30], int acontactNumber, const char aemail[50], const char
aAdress[100], int accID, const char nicNo[12]);

    void updateAccountant(int accID);

    void calculateSalary();

    void calculatePayment();

    ~Accountant()

    {

        cout << "Accountant " << accID << "deleted" << endl;

    };

};

```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21215810
```

```
//class Admin
```

```
#include <iostream>
```

```
#include <cstring>
```

```
#include "Feedback.h"
```

```
#include "Employee.h"
```

```
using namespace std;
```

```
class Admin : public Employee{
```

```
protected:
```

```
    int adminID;
```

```
    Feedback* Fb[5];
```

```
public :
```

```
    Admin()
```

```
{
```

```
    strcpy(name, "");
```

```
    contactNumber = 0;
```

```
    strcpy(email, "");
```

```
    strcpy(adress, "");
```

```
    strcpy(position, "");
```

```
    strcpy(NIC, "");
```

```
    adminID = 0;
```

```
};
```

```
char replyFeedback();
```

```
void viewFeedback();
```

```
~Admin()
```

```
{
```

```
    cout << "Admin " << adminID << "deleted " << endl ;
```

```
};
```

```
};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21215810
```

```
//Driver class
```

```
#include <iostream>
```

```
#include <cstring>
```

```
#include "Employee.h"
```

```
using namespace std;
```

```
class Driver: public Employee{
```

```
protected:
```

```
    int dID;
```

```
    int licenseNo;
```

```
public:
```

```
    Driver()
```

```
    {
```

```
        strcpy(name, "");
```

```
        contactNumber = 0;
```

```
        strcpy(email, "");
```

```
        strcpy(adress, "");
```

```
        strcpy(position, "");
```

```
        strcpy(NIC, "");
```

```
        dID = 0;
```

```
        licenseNo = 0;
```



```
};
```

```
void adddriver(const char dname[30], int dcontactNumber, const char demail[50],const char  
dadress[100], int dID, const char nicNo[12]) ;
```

```
void updatedriver(int dID,int licenseno);
```

```
void requestdriver(int dID,int licenseno);
```

```
~Driver()
```

```
{
```

```
    cout << "Driver " << dID << " deleted " << endl;
```

```
};
```

```
};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21214516
```

```
//Class Payment
```

```
#include <iostream>
```

```
#include "Discount.h"
```

```
using namespace std;
```

```
class Payment{
```

```
private:
```

```
    int paymentID;
```

```
    double amount;
```

```
    char type[10];
```

```
    Discount *dis;
```

```
public:
```

```
    Payment()
```

```
{
```

```
    dis = new Discount (001, 520.00);
```

```
};
```

```
Payment(int ID, double dAmount)
```

```
{
```

```
    dis = new Discount (int ID, double dAmount);
```

```
};
```

```
void makePayment(int paymentID, double amount,const char type[10]);
```

```
void payPayment();
```

```
~Payment()
```

```
{
```

```
    delete dis;
```

```
    cout << "Discount is  deleted" << endl;
```

```
};
```

```
};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21214516
```

```
//Discount class
```

```
#include <iostream>
```

```
using namespace std;
```

```
class Discount{
```

```
private:
```

```
    int discountID;
```

```
    double disAmount;
```

```
public:
```

```
    Discount()
```

```
{
```

```
    discountID = 0;
```

```
    disAmount = 0.0;
```

```
};
```

```
    Discount(int dID, double Amount)
```

```
{
```

```
    discountID = dID;
```

```
    disAmount = Amount;
```

```
};
```

```
    void prevReservation();
```

```
    double calDiscount(double payment);
```

```
void addDiscount();
```

```
~Discount()
```

```
{
```

```
    cout << "Discount " << discountID << " deleted " << endl;
```

```
};
```

```
};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21214516
```

```
//Vehicle class
```

```
#include <iostream>
```

```
#include <cstring>
```

```
#include "User.h"
```

```
using namespace std;
```

```
class Vehicle
```

```
{
```

```
    private:
```

```
        char vehicleType[10];
```

```
        int registrationNo;
```

```
        char color[10];
```

```
        double rentalCharge;
```

```
        User* U[5];
```

```
    public:
```

```
        Vehicle()
```

```
        {
```

```
            strcpy(vehicleType, "");
```

```
            registrationNo = 0;
```

```
            strcpy(color, "");
```

```
            rentalCharge = 0.0;
```

```
        };
```

```
void addVehicleDetails(const char vType[10], const char regNo, const char vColor[10], double  
rCharge);
```

```
void updateVehicleType(const char vType[10], const char regNo, const char vColor[10], double  
rCharge);
```

```
void checkAvailability();
```

```
~Vehicle()
```

```
{
```

```
    cout << "Vehicle " << registrationNo << " deleted " << endl;
```

```
};
```

```
};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21216596
```

```
//class Reservation
```

```
#include <iostream>
```

```
#include <cstring>
```

```
#include "User.h"
```

```
using namespace std;
```

```
class Reservation
```

```
{
```

```
    private:
```

```
        int rID;
```

```
        char pickupDate[10];
```

```
        char pickupTime[8];
```

```
        char pickupLocation[30];
```

```
        char vehicle[15];
```

```
        int rentalDuration;
```

```
        int noofPassengers;
```

```
        User* u[5];
```

```
    public:
```

```
        Reservation()
```

```
        {
```

```
            rID = 0;
```

```
            strcpy(pickupDate, "");
```

```
            strcpy(pickupTime, " ");
```

```
            strcpy(pickupLocation, "");
```

```
            strcpy(vehicle, "");
```



```
    rentalDuration = 0;

    noofPassengers = 0;

};
```

```
    void makeReservation(int rID, const char pDate[10], const char pTime[8], const char
pLocation[30], const char vName[15], int rDuration, int Passengers);
```

```
void cancelReservation();
```

```
~Reservation()
```

```
{

    cout << "Reservation " << rID << " deleted "<<endl;

};
```

```
};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21214820
```

```
//Feedback class
```

```
#include <iostream>
```

```
#include <cstring>
```

```
using namespace std;
```

```
class Feedback{
```

```
private:
```

```
    int feedbackID;
```

```
    char fbDescription;
```

```
    char fbDate;
```

```
public:
```

```
    Feedback()
```

```
    {
```

```
        feedbackID = 0;
```

```
        strcpy(fbDescription, "");
```

```
        strcpy(fbDate, "");
```

```
    };
```

```
void sendFeedback(int fbID, const char fbDescrip[100], const char Date[10]);
```

```
void removeFeedback();
```

```
void updateFeedback(int fbID, const char fbDescrip[100], const char Date[10]);
```

```
~Feedback()

{
    cout << "Feedback " << feedbackID << " deleted " << endl;
};

};
```

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21216596
```

```
//class report
```

```
#include<iostream>
```

```
using namespace std;
```

```
class Report{
```

```
private:
```

```
    int reportID;
```

```
public:
```

```
    Report()
```

```
{
```

```
    reportID = 0;
```

```
};
```

```
    Report( int reportID);
```

```
    void genaratereport(int reportID);
```

```
    ~Report()
```

```
{
```

```
    cout << "Report " << reportID<< " deleted " <<endl;
```

```
};
```

```
};
```

## Main program

```
//MLB_02.02_05_VehicleRentalSystem
```

```
//IT21214820
```

```
//main program
```

```
//Including headerfiles of classes
```

```
#include "Accountant.h"
```

```
#include "Employee.h"
```

```
#include "User.h"
```

```
#include "Admin.h"
```

```
#include "Discount.h"
```

```
#include "Driver.h"
```

```
#include "Feedback.h"
```

```
#include "Payment.h"
```

```
#include "Report.h"
```

```
#include "Reservation.h"
```

```
#include "Vehicle.h"
```

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    //Creating dynamic objects of User class
```

```
    class User* U1 = new User();
```

```
    U1->addUserDetails("Ajith", 071420000, "Ajith@gmail.com","Colombo");
```

```
    delete U1;
```

//Creating dynamic object of Employee class

```
Employee* E1 = new Employee();
```

```
E1->addEmployee("Sujith",012365405, "sujith@gmail.com", "Kegalle", 001, "Receptionist",  
"1615280423v");
```

```
delete E1;
```

//Creating dynamic object of Accountant class

```
Accountant* Ac1 = new Accountant();
```

```
Ac1->addAccountant("Kamal", 01236541, "Kamal@gmail.com",  
"Anuradhapura",101,"1212364v");
```

```
delete Ac1;
```

//Creating dynamic object of Admin class

```
Admin* Ad1 = new Admin();
```

```
Ad1->viewFeedback();
```

```
delete Ad1;
```

//Creating dynamic object of Driver class

```
Driver* D1 = new Driver();
```

```
D1->updatedriver(101,12645236);
```

```
delete D1;
```

//Creating dynamic object of Reservation class

```
Reservation* Res1 = new Reservation();
```

```
Res1->makeReservation(10235, "19 June 2022", "8.30am", "Colombo", "Cab no123 Ac", 2, 1);
```

```
delete Res1;
```

//Creating dynamic object of Payment class

Payment\* Pay1= new Payment();

Pay1->makePayment(123, 45000.00, "Credit card");

delete Pay1;

//Creating dynamic object for Vehicle class

Vehicle\* V1 = new Vehicle();

V1->addVehicleDetails("Car", "1234AC", "Black", 1500.00);

delete V1;

//Creating dynamic object for Report class

Report\* Rep1 = new Report();

Rep1->genaratereport(321);

delete Rep1;

//Creating dynamic object for Feedback class

Feedback\* Fb1 = new Feedback();

Fb1->sendFeedback(456, "Good service", "16 February 2022");

delete Fb1;

return 0;

}

## Group Contribution

Student ID	Name	Contribution
IT21214820	Dharmathilake K.A.D.K.D.	<ul style="list-style-type: none"><li>• Coding User class</li><li>• Coding Feedback class</li><li>• Coding main program</li></ul>
IT21214516	Fernando N.K.B	<ul style="list-style-type: none"><li>• Coding Vehicle class</li><li>• Coding Payment class</li><li>• Coding Discount class</li></ul>
IT21213144	Senarathne D.R.	<ul style="list-style-type: none"><li>• Coding Employee class</li><li>• Coding Accountant</li></ul>
IT21215810	Pathirana G.P.M.P.	<ul style="list-style-type: none"><li>• Coding Driver class</li><li>• Coding Admin class</li></ul>
IT21216596	Thihansa K.A.S.	<ul style="list-style-type: none"><li>• Coding Reservation class</li><li>• Coding Report class</li></ul>