UNIVERSITY OF WESTMINSTER

INFORMATICS INSTITUTE OF TECHNOLOGY

INFORMATICS INSTITUTE OF TECHNOLOGY

IN COLLABORATION WITH

UNIVERSITY OF WESTMINSTER (UOW)

BENG (HONS) SOFTWARE ENGINEERING

MODULE: 5COSCOO1W Object Oriented Programming

MODULE LEADER : MR. P.GUGANATHAN

COURSE WORK – 01

UOW ID: W16541915/1

STUDENT ID: 2016323

STUDENT FIRST NAME: KAJENDRAN

STUDENT SURNAME: CHANDRESWARAN

Table of contents

Acknowledgement

I would like to thank Mr. P. Guganathan for the guidance and immense support given towards this Object Oriented Programming module and tolerating all the time.

Finally I am really thankful to my parents and my friends for being patient and encouraging me continuously.

Introduction

This program is about to register the vehicles, monitor them, calculate the cost of the parking, Viewing the parking area of the vehicles and finally removing the vehicle when it goes out. The parking area contains 20 parking slots. After adding and removing vehicle program should show the balance units available in parking area.

Requirements

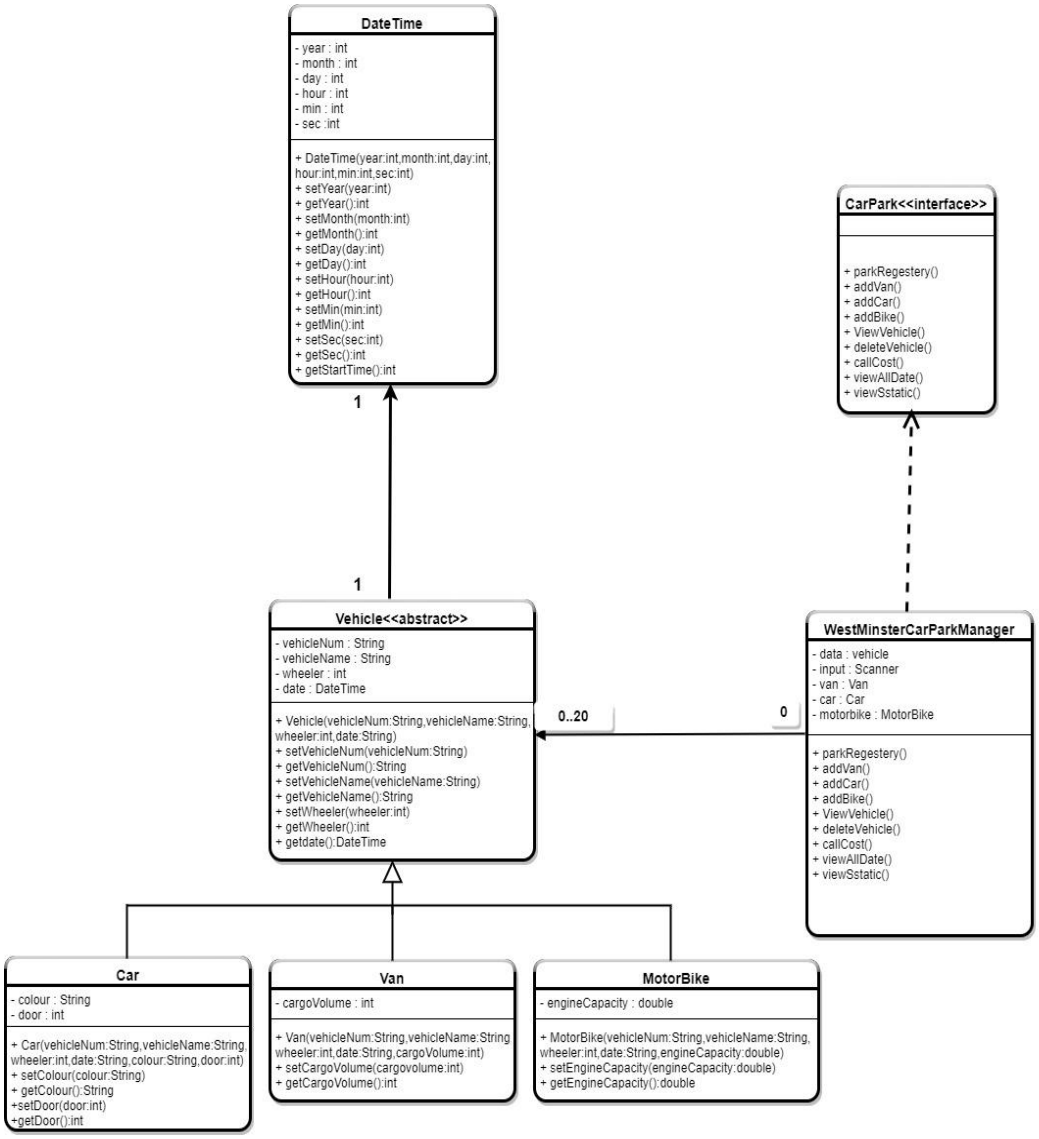
Functional requirements

1- Add vehicle: User should able to add the vehicle (van, car, motorbike) program should consider about the available parking slots in the area while adding the vehicles.
2- View vehicles: User should able view all the vehicle at the time.
3- Remove vehicle: User should able to remove the vehicles when it departs from the car parking.
4- Calculate bill: User should able make a bill calculation according to their duration of parking.
5- View all data: User can be able to view all the data of the date.

Non-Functional requirements

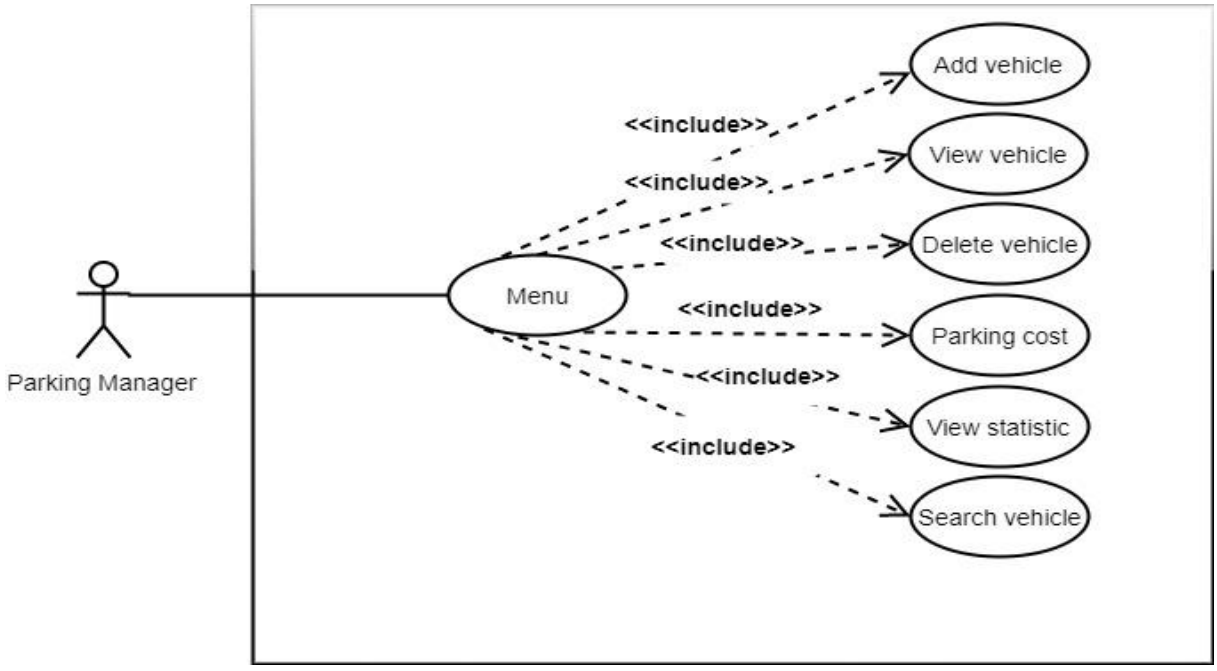
1- Create a user interface (menu and easy to access).

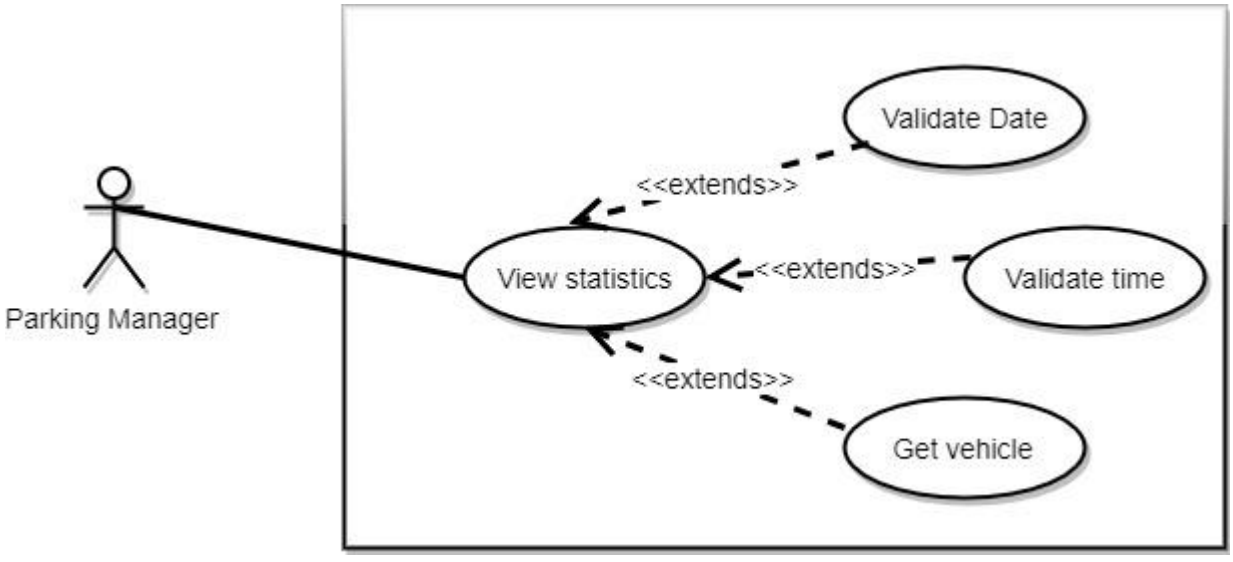
Class diagram



Use

Use case diagram





Blackbox testing

Test number	Test description	Expected result	Actual Result	Result
01	Test the menu. Press 1 to register the vehicle. Press 3 to remove the vehicle. Press 0 to view the statics of vehicle	Display vehicle adding menu Display the remove menu. Display the statics	Displayed the vehicle adding menu Displayed the remove menu. displayed statics data	Pass Pass Pass
02	Test the Date and time. Enter date and time 11.5.2017.11.11.12 this format.	Date and time will added to the vehicle	date and time added	Pass
03	Add only one vehicle and try to view it	Display the detail of the vehicle	Displayed the data of the vehicle.	Pass
04	Test calculating the statics. Add one vehicle per every vehicles and try to view the statics	Display the statics	Displayed statics	Pass

White box testing

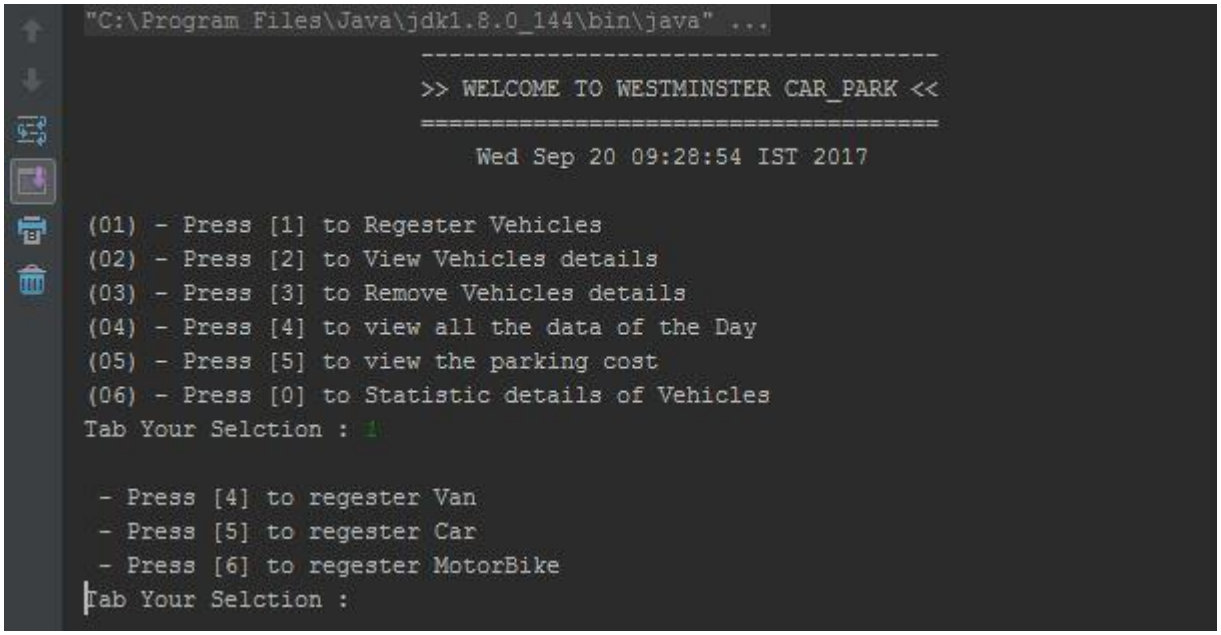
Test number	Test description	Expected output	Actual output	Result
01	Try to view statics before adding anydata	Display error message	Displayed error message	Pass
02	Insert date and time in 11*12*2017*14*25*56	Display an error message	Displayed an error message	Pass
03	Try to view vehicle before adding data	Display an error message	Display an error message	Pass
04	input 78 in the main menu	Display an error message	Displayed an error message	Pass

Codes and Screenshots

1 - Display menu

```
public static void parkRegestry() {
    System.out.println("\n" + "(01) - Press [1] to Regester Vehicles  ");
    System.out.println("(02) - Press [2] to View Vehicles details ");
    System.out.println("(03) - Press [3] to Remove Vehicles details ");
    System.out.println("(04) - Press [4] to view all the data of the Day ");
    System.out.println("(05) - Press [5] to view the parking cost ");
    System.out.println("(06) - Press [0] to Statistic details of Vehicles ");
    System.out.print("Tab Your Selction : ");
    String user = input.next();

    switch (user) {
        case "1":
            System.out.println("\n" + " - Press [4] to regester Van");
            System.out.println(" - Press [5] to regester Car");
            System.out.println(" - Press [6] to regester MotorBike");
            System.out.print("Tab Your Selction : ");
            String user1 = input.next();
            switch ((user1)) {
                case "4":
                    parkVan();
                    break;
```

```
                case "5":
                    parkCar();
                    break;
                case "6":
                    parkBike();
                    break;
                default:
                    System.err.println("Iinvalid selection " + "\n");
                    parkRegestry();
            }

            break;
        case "2":
            if (list.isEmpty() && list1.isEmpty() && list2.isEmpty()) {
                System.err.println(" There is no details , Please Enter details ");
                parkRegestry();
            } else {
                viewVan();
                viewCar();
                viewBike();
            }
            break;
        case "3":
            System.out.println("\n" + " - Press [v] to release Van");
            System.out.println(" - Press [c] to release Car");
            System.out.println(" - Press [m] to release MotorBike");
            System.out.print("Tab Your Selction : ");
            user1 = input.next();
            switch ((user1)) {
                case "v":
                    releasePark();
                    break;
                case "c":
                    releasePark1();
                    break;
                case "m":
                    releasePark2();
                    ;
                    break;
            }
       case "4":

            viewAllData();

            break;

        case "5":

            calCost();

            break;

        case "0":

            viewStatic();

            break;

        default:

            System.err.println("invalid input Please read the instruction and , try
again");
```



2 - Add Vehicle

```
public static void parkVan() {
    System.out.println("\n" + "            Regester the Van details in oder");
    System.out.println("            ---------------------------------");
    System.out.print("[01] - Enter Vehicle Name : ");
    String vehicleName = input.next();
    System.out.print("[02] - Enter Vehicle number : ");
    String number = input.next();
    System.out.print("[03] - Enter the wheeler : ");
    int wheeler = input.nextInt();
    System.out.print("[04] - Enter the Cargo volume : ");
    int volume = input.nextInt();
    System.out.print("[04] - Enter the Date and time (DD.MM.YYYY.HH.MM.SS) : ");
    String date = input.next();
    String[] day = date.split("\\.");
    --slots;
    list.add(new Van(vehicleName, number, wheeler, volume, date));
    list3.add((new Abstract(vehicleName, number, wheeler, date)));
    end();
```

```
  - Press [4] to regester Van
  - Press [5] to regester Car
  - Press [6] to regester MotorBike
 Tab Your Selction : 4

             Regester the Van details in oder
             --------------------------------
 [01] - Enter Vehicle Name : Toyato
 [02] - Enter Vehicle number : csk1111
 [03] - Enter the wheeler : 4
 [04] - Enter the Cargo volume : 14
 [04] - Enter the Date and time (DD.MM.YYYY.HH.MM.SS) : 11.5.2017.12.12.12

 Press [9] to END
 Press [8] to COUNTINUE
 Tab Your Selction :
```

3 - View Vehicle

```
public static void viewVan() {
    Collections.reverse(list);
    for (int i = 0; i < list.size(); i++) {
        System.out.print("\n" + "Date of Enter : " + list.get(i).getDate() + "\n");
        System.out.println("Entered Time : " + list.get(i).getDate().toTime());
        System.out.print("Vehile type : " + list.get(i).getVehicleName() + "\n");
        System.out.print("Vehile number : " + list.get(i).getVehicleNum() + "\n");}
public static void viewCar() {
    Collections.reverse(list1);
    for (int i = 0; i < list1.size(); i++) {
        System.out.print("\n" + "Date of Enter : " +
list1.get(i).getDate().toString() + "\n");
        System.out.println("Entered Time : " + list1.get(i).getDate().toTime());
        System.out.print("Vehile type : " + list1.get(i).getVehicleName() + "\n");
        System.out.print("Vehile number : " + list1.get(i).getVehicleNum() + "\n");
public static void viewBike() {
    Collections.reverse(list2);
    for (int i = 0; i < list2.size(); i++) {
        System.out.print("\n" + "Date of Enter : " +
list2.get(i).getDate().toString() + "\n");
        System.out.println("Entered Time : " + list2.get(i).getDate().toTime());
        System.out.print("Vehile type : " + list2.get(i).getVehicleName() + "\n");
        System.out.print("Vehile number : " + list2.get(i).getVehicleNum() + "\n");
        System.out.print("Engine Capacity : " + list2.get(i).getEngineCapacity() +
"\n");}
```

```
System.out.print("\nAvailable parking slots : " + slots + "\n");end();}
```

```
(01) - Press [1] to Regester Vehicles
(02) - Press [2] to View Vehicles details
(03) - Press [3] to Remove Vehicles details
(04) - Press [4] to view all the data of the Day
(05) - Press [5] to view the parking cost
(06) - Press [6] to find vehicle by date
(06) - Press [0] to Statistic details of Vehicles
Tab Your Selction : 2

Date of Enter : 2017:5:11
Entered Time : 12:12:12
Vehile type : toyato
Vehile number : csk1111

Date of Enter : 2017:5:11
Entered Time : 12:13:15
Vehile type : nissan
Vehile number : csp1212

Date of Enter : 2017:5:11
Entered Time : 1:15:14
Vehile type : Pulsar
Vehile number : csl1212
Engine Capacity : 1500.0

Available parking slots : 17
```

4 - Remove vehicle

```
public static void releasePark() {
    System.out.println("No of Vans in the parking : " + list.size());
    for (int x = 0; x < list.size(); x++) {
        System.out.println("Plate Id of the vehicle : " +
list.get(x).getVehicleNum());
    }
    System.out.print("Enter Vehicle plat Id to remove :");
    String carName = input.next();// insert code here
    int index = -1;
    for (int i = 0; i < list.size(); i++) {
        if (list.get(i).getVehicleNum().equals(carName)) {
            index = i;
            System.out.println("Vehile number : " + list.get(i).getVehicleNum());
            System.out.println("Entered Date : " +
list.get(i).getDate().toString());
            list.remove(list.remove(i));
        }
    }
    System.out.print("Available parking slots : " + slots + "\n");
```

```
    end();}
```

```
(01) - Press [1] to Regester Vehicles
(02) - Press [2] to View Vehicles details
(03) - Press [3] to Remove Vehicles details
(04) - Press [4] to view all the data of the Day
(05) - Press [5] to view the parking cost
(06) - Press [6] to find vehicle by date
(06) - Press [0] to Statistic details of Vehicles
Tab Your Selction : 3

 - Press [v] to release Van
 - Press [c] to release Car
 - Press [m] to release MotorBike
Tab Your Selction : v
No of Vans in the parking : 1
Plate Id of the vehicle : csp1212
Enter Vehicle plat Id to remove :csp1212

Vehile number : csp1212
Entered Date : 2017:5:11
```

5- View statics

```
public static void viewStatic() {

    double sVan = list.size();

    System.out.println("\n - The percentage of Vans in the parking : " + ((sVan *
100) / 20) + "%");

    double sCar = list1.size();

    System.out.println(" - The percentage of Cars in the parking : " + ((sCar * 100)
/ 20) + "%");

    double sBike = list2.size();

    System.out.println(" - The percentage of Bikes in the parking : " + ((sBike *
100) / 20) + "%");

    System.out.println(" - The total filled space of the parking : " + (((sVan *
100) / 20)+((sCar * 100) / 20)+((sBike * 100) / 20)) + "%");    end();}
```

```
(01) - Press [1] to Regester Vehicles
(02) - Press [2] to View Vehicles details
(03) - Press [3] to Remove Vehicles details
(04) - Press [4] to view all the data of the Day
(05) - Press [5] to view the parking cost
(06) - Press [6] to find vehicle by date
(06) - Press [0] to Statistic details of Vehicles
Tab Your Selction : 0

 - The percentage of Vans in the parking : 5.0%
 - The percentage of Cars in the parking : 5.0%
 - The percentage of Bikes in the parking : 0.0%
 - The total filled space of the parking : 10.0%
```

6 - View Bill cost

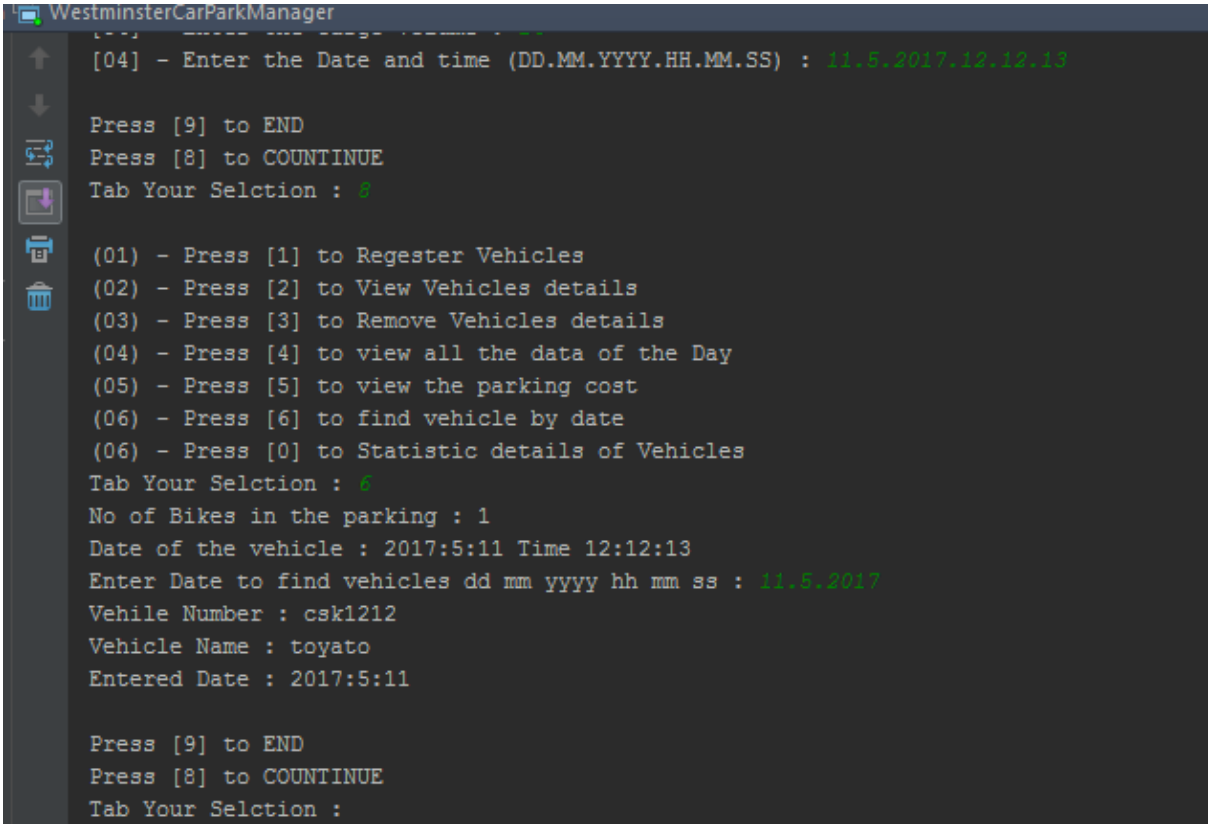
```
public static void calCost(){
   System.out.println("Enter Leaving time : ");
   String lDate = input.next();
   String[] day = lDate.split("\\.");
   int hour = Integer.parseInt(day[3]);
   int min = Integer.parseInt(day[4]);
   int sec = Integer.parseInt(day[5]);
   int endTime = ((hour*60*60)+(min*60)+sec);
   for(int x=0; x<list3.size(); x++) {
       System.out.println("Vehile number : " + list3.get(x).getVehicleNum());
       System.out.println("Entered Date : " + list3.get(x).getDate().toString());
       double cost = 0.0;
       int hour1 = (endTime - (list3.get(x).getDate().getStartTime())) / 3600;
       if (hour1 <= 3) {
           cost = hour1 * 3;
           System.out.println("The cost is 3$ per hour : " + cost + "$");
       } else if(hour1 <= 24){
           cost = (hour1 - 3) + 9;
           System.out.println("The cost is 9$ for first 3hour and 1$ per hour : " +
cost + "$"); }}
```

```
(01) - Press [1] to Regester Vehicles
(02) - Press [2] to View Vehicles details
(03) - Press [3] to Remove Vehicles details
(04) - Press [4] to view all the data of the Day
(05) - Press [5] to view the parking cost
(06) - Press [6] to find vehicle by date
(06) - Press [0] to Statistic details of Vehicles
Tab Your Selction : 5
Enter Leaving time :
11.5.2017.20.20.20
Vehile number : csp1542
Entered Date : 2017:5:11
The cost is 9$ for first 3hour and 1$ per hour : 11.0$
Vehile number : cdp9689
Entered Date : 2017:5:11
The cost is 9$ for first 3hour and 1$ per hour : 11.0$
```

7 - View vehicles by date

```
public static void findVehicle() {

   System.out.println("No of Bikes in the parking : " + list3.size());

   for (int x = 0; x < list3.size(); x++) {

       System.out.println("Date of the vehicle : " +
list3.get(x).getDate().toString() + " Time " +  list3.get(x).getDate().toTime());

   }

   System.out.print("Enter Date to find vehicles dd mm yyyy hh mm ss : ");

   String date = input.next();// insert code here

   String []day = date.split("\\.");

   //int index = -1;

   int year = Integer.parseInt(day[2]);

   int month = Integer.parseInt(day[1]);

   int toDay = Integer.parseInt(day[0]);

   for (int i = 0; i < list3.size(); i++) {

       if (list3.get(i).getDate().getDay() == year &&
list3.get(i).getDate().getMonth() == month && list3.get(i).getDate().getYear() ==
toDay) {

           System.out.println("Vehile Number : " + list3.get(i).getVehicleNum());

           System.out.println("Vehicle Name : " + list3.get(i).getVehicleName());

           System.out.println("Entered Date : " +
list3.get(i).getDate().toString());

       }
```

```
    }
```




```
    end();}
```

Concluation

This coursework let me to improve my programming skills in,

1 - How to implement the OOP principles in the programming.

2 - How to handle the error handling.

3 - How OOP concepts help to reduce the code , reuseablity of coding.

4 - How to do efficient coding.

5 - How to secure our coding.

- This course work taught me how to do a proper course work in future, analyzing the requirements, planning the structure, designing the program, validating, testing, how to write a report to a course work and time management.