INFORMATICS INSTITUTE OF TECHNOLOGY

UNIVERSITY OF WESTMINSTER

Informatics Institute of Technology

Department of Computing

(B.Sc.) in Computer Science

Module: 5COSC007C.1

Object Oriented Programming

OOP Coursework

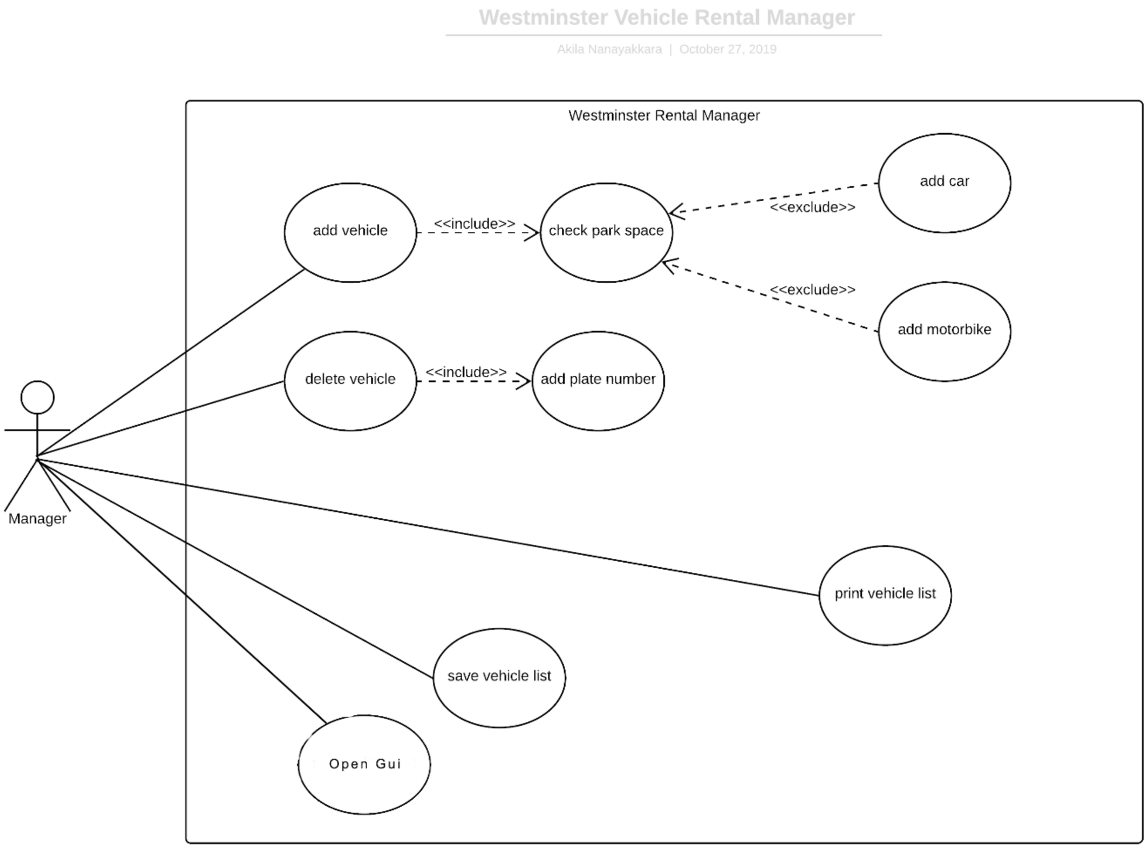
Date	: 02/12/2019
Student ID	: 2018400
Student UoW ID	: w1742308
Student First Name	: Akila
Student Surname	: Nanayakakra
Tutorial Group	: Group B

Table of Contents

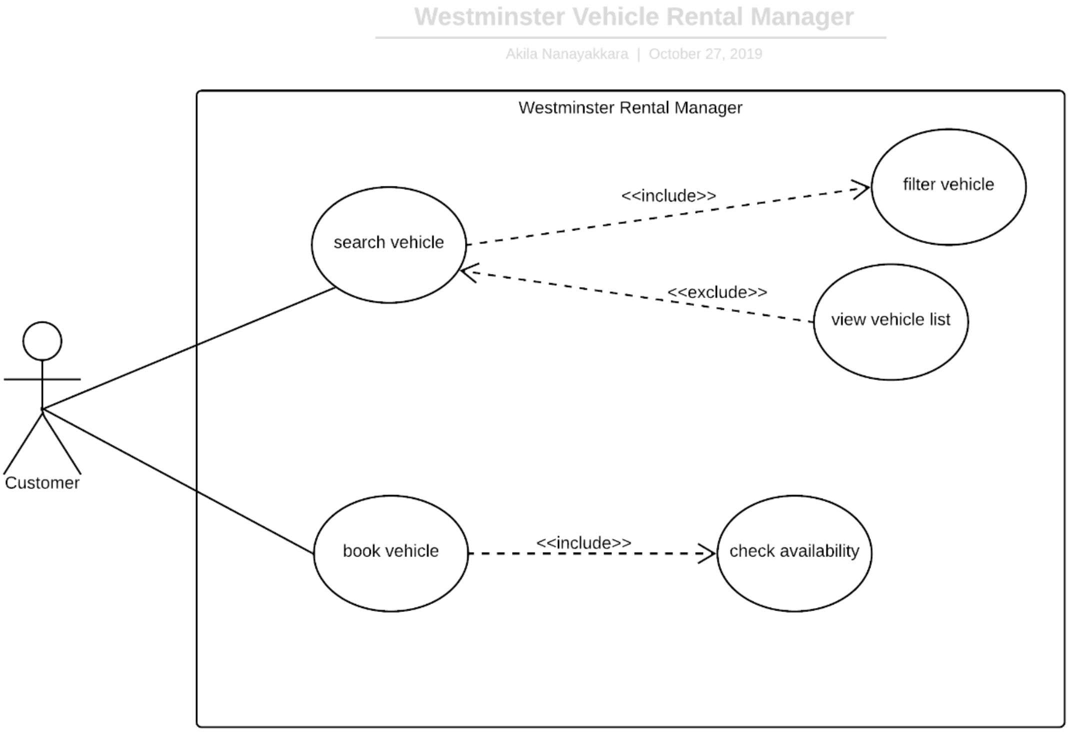
❖ Design and Solution

○ Use Cases

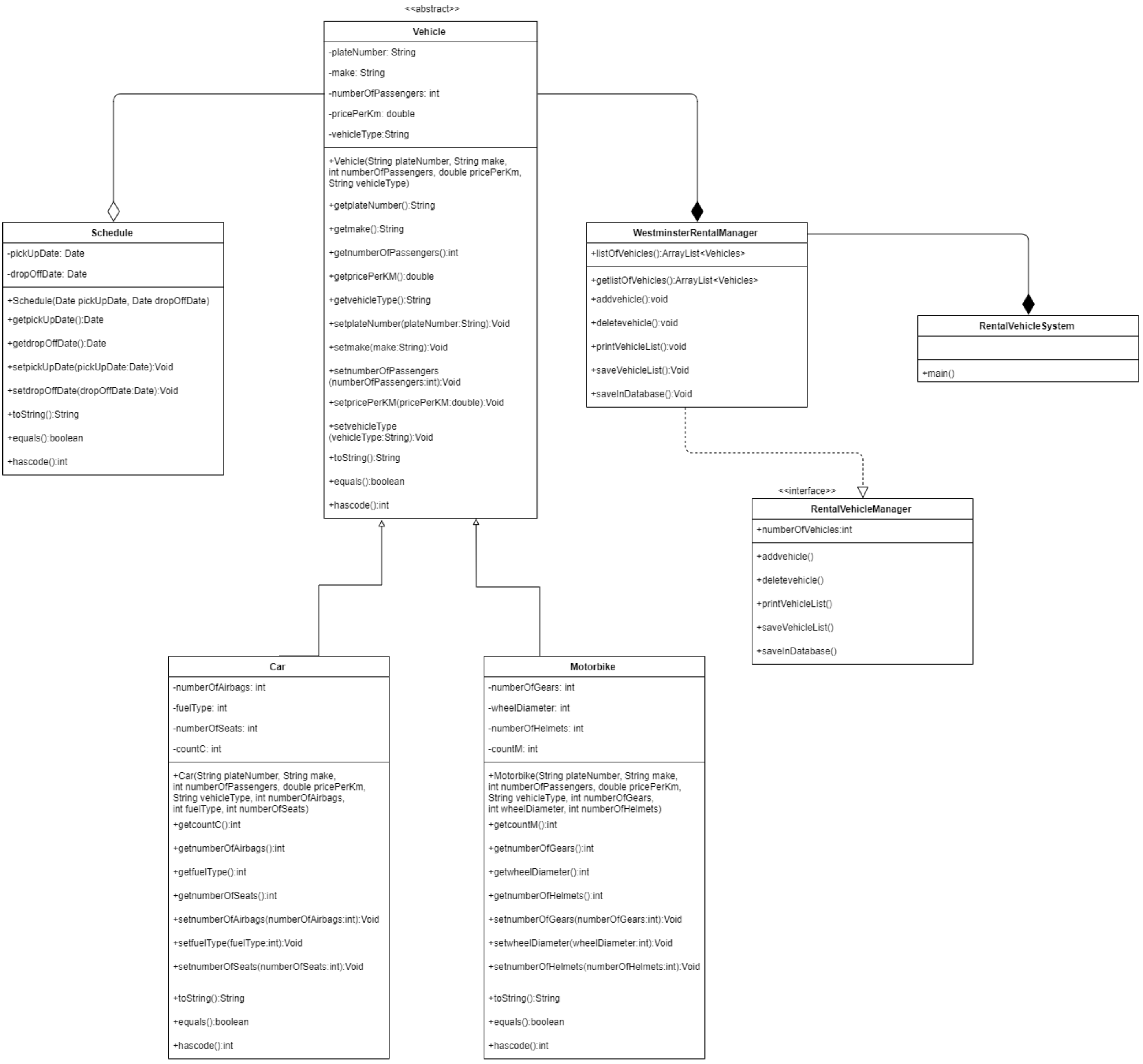
Use case for the console application



Use case for the gui application



- Class Diagram


<<abstract>>
Vehicle
-plateNumber: String
-make: String
-numberOfPassengers: int
-pricePerKm: double
-vehicleType:String
+Vehicle(String plateNumber, String make, int numberOfPassengers, double pricePerKm, String vehicleType)
+getplateNumber():String
+getmake():String
+getnumberOfPassengers():int
+getpricePerKM():double
+getvehicleType():String
+setplateNumber(plateNumber:String):Void
+setmake(make:String):Void
+setnumberOfPassengers (numberOfPassengers:int):Void
+setpricePerKM(pricePerKM:double):Void
+setvehicleType (vehicleType:String):Void
+toString():String
+equals():boolean
+hascode():int
Schedule
-pickUpDate: Date
-dropOffDate: Date
+Schedule(Date pickUpDate, Date dropOffDate)
+getpickUpDate():Date
+getdropOffDate():Date
+setpickUpDate(pickUpDate:Date):Void
+setdropOffDate(dropOffDate:Date):Void
+toString():String
+equals():boolean
+hascode():int
WestminsterRentalManager
+listOfVehicles():ArrayList<Vehicles>
+getlistOfVehicles():ArrayList<Vehicles>
+addvehicle():void
+deletevehicle():void
+printVehicleList():void
+saveVehicleList():Void
+saveInDatabase():Void
RentalVehicleSystem
+main()
<<interface>>
RentalVehicleManager
+numberOfVehicles:int
+addvehicle()
+deletevehicle()
+printVehicleList()
+saveVehicleList()
+saveInDatabase()
Car
-numberOfAirbags: int
-fuelType: int
-numberOfSeats: int
-countC: int
+Car(String plateNumber, String make, int numberOfPassengers, double pricePerKm, String vehicleType, int numberOfAirbags, int fuelType, int numberOfSeats)
+getcountC():int
+getnumberOfAirbags():int
+getfuelType():int
+getnumberOfSeats():int
+setnumberOfAirbags(numberOfAirbags:int):Void
+setfuelType(fuelType:int):Void
+setnumberOfSeats(numberOfSeats:int):Void
+toString():String
+equals():boolean
+hascode():int
Motorbike
-numberOfGears: int
-wheelDiameter: int
-numberOfHelmets: int
-countM: int
+Motorbike(String plateNumber, String make, int numberOfPassengers, double pricePerKm, String vehicleType, int numberOfGears, int wheelDiameter, int numberOfHelmets)
+getcountM():int
+getnumberOfGears():int
+getwheelDiameter():int
+getnumberOfHelmets():int
+setnumberOfGears(numberOfGears:int):Void
+setwheelDiameter(wheelDiameter:int):Void
+setnumberOfHelmets(numberOfHelmets:int):Void
+toString():String
+equals():boolean
+hascode():int

❖ Implementing functionality

- ### Vehicle Class

```java
package lk.oopCoursework1;

import java.util.Objects;

public abstract class Vehicle {
    private String plateNumber;
    private String make;
    private int numberOfPassengers; //for both motor bike and car
    private double pricePerKm;//for both motor bike and car
    private String vehicleType;
    public Vehicle(){}

    public Vehicle(String plateNumber, String make, int numberOfPassengers, double
pricePerKm, String vehicleType) {
        this.plateNumber = plateNumber;
        this.make = make;
        this.numberOfPassengers = numberOfPassengers;
        this.pricePerKm = pricePerKm;
        this.vehicleType = vehicleType;
    }

    public String getPlateNumber() {
        return plateNumber;
    }

    public void setPlateNumber(String plateNumber) {
        this.plateNumber = plateNumber;
    }

    public String getMake() {
        return make;
    }

    public void setMake(String make) {
        this.make = make;
    }

    public int getNumberOfPassengers() {
        return numberOfPassengers;
    }

    public void setNumberOfPassengers(int numberOfPassengers) {
        this.numberOfPassengers = numberOfPassengers;
    }

    public double getPricePerKm() {
        return pricePerKm;
    }

    public void setPricePerKm(double pricePerKm) {
        this.pricePerKm = pricePerKm;
    }

    public String getVehicleType() {
        return vehicleType;
    }
    public void setVehicleType(String vehicleType) {
        this.vehicleType = vehicleType;
    }

    @Override
    public String toString() {
        return "Vehicle{" +
                "plateNumber='" + plateNumber + '\'' +
                ", make='" + make + '\'' +
                ", numberOfPassengers=" + numberOfPassengers +
                ", pricePerKm=" + pricePerKm +
                ", vehicleType='" + vehicleType + '\'' +
                '}';
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Vehicle vehicle = (Vehicle) o;
        return numberOfPassengers == vehicle.numberOfPassengers &&
                Double.compare(vehicle.pricePerKm, pricePerKm) == 0 &&
                Objects.equals(plateNumber, vehicle.plateNumber) &&
                Objects.equals(make, vehicle.make) &&
                Objects.equals(vehicleType, vehicle.vehicleType);
    }

    @Override
    public int hashCode() {
        return Objects.hash(plateNumber, make, numberOfPassengers, pricePerKm,
vehicleType);
    }
}
```

- Car Class

```
package lk.oopCoursework1;

import java.util.Objects;

public class Car extends Vehicle{
    private int numberOfAirbags;
    private int fuelType;
    private int numberOfSeats;
    private static int countC =0;

    public Car(String plateNumber, String make, int numberOfPassengers, double
pricePerKm, String vehicleType, int numberOfAirbags, int fuelType, int
numberOfSeats) {
        super(plateNumber, make, numberOfPassengers, pricePerKm, vehicleType);
        this.numberOfAirbags = numberOfAirbags;
        this.fuelType = fuelType;
        this.numberOfSeats = numberOfSeats;
        countC +=1;
    }

    public Car() {
    }
    public static int getCountC() {
        return countC;
    }

    public int getNumberOfAirbags() {
        return numberOfAirbags;
    }

    public void setNumberOfAirbags(int numberOfAirbags) {
        this.numberOfAirbags = numberOfAirbags;
    }

    public int getFuelType() {
        return fuelType;
    }

    public void setFuelType(int fuelType) {
        this.fuelType = fuelType;
    }

    public int getNumberOfSeats() {
        return numberOfSeats;
    }

    public void setNumberOfSeats(int numberOfSeats) {
        this.numberOfSeats = numberOfSeats;
    }

    @Override
    public String toString() {
        return "Car{" +
                "numberOfAirbags=" + numberOfAirbags +
                ", fuelType=" + fuelType +
                ", numberOfSeats=" + numberOfSeats +
                "} " + super.toString();
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        if (!super.equals(o)) return false;
        Car car = (Car) o;
        return numberOfAirbags == car.numberOfAirbags &&
                fuelType == car.fuelType &&
                numberOfSeats == car.numberOfSeats;
    }

    @Override
    public int hashCode() {
        return Objects.hash(super.hashCode(), numberOfAirbags, fuelType,
numberOfSeats);
    }
}
```

- Motorbike Class

```
package lk.oopCoursework1;

import java.util.Objects;

public class Motorbike extends Vehicle{

    private int numberOfGears;
    private int wheelDiameter;
    private int numberOfHelmets;
    private static int countM =0;

    public Motorbike(String plateNumber, String make, int numberOfPassengers, double
pricePerKm, String vehicleType, int numberOfGears, int wheelDiameter, int
numberOfHelmets) {
        super(plateNumber, make, numberOfPassengers, pricePerKm, vehicleType);
        this.numberOfGears = numberOfGears;
        this.wheelDiameter = wheelDiameter;
        this.numberOfHelmets = numberOfHelmets;
        countM +=1;
    }

    public Motorbike() {

    }

    public static int getCountM() {
        return countM;
    }

    public int getNumberOfGears() {
        return numberOfGears;
    }

    public void setNumberOfGears(int numberOfGears) {
        this.numberOfGears = numberOfGears;
    }

    public int getWheelDiameter() {
        return wheelDiameter;
    }

    public void setWheelDiameter(int wheelDiameter) {
        this.wheelDiameter = wheelDiameter;
    }

    public int getNumberOfHelmets() {
        return numberOfHelmets;
    }

    public void setNumberOfHelmets(int numberOfHelmets) {
        this.numberOfHelmets = numberOfHelmets;
    }

    @Override
    public String toString() {
        return "Motorbike{" +
                "numberOfGears=" + numberOfGears +
                ", wheelDiameter=" + wheelDiameter +
                ", numberOfHelmets=" + numberOfHelmets +
                "} " + super.toString();
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        if (!super.equals(o)) return false;
        Motorbike motorbike = (Motorbike) o;
        return numberOfGears == motorbike.numberOfGears &&
                wheelDiameter == motorbike.wheelDiameter &&
                numberOfHelmets == motorbike.numberOfHelmets;
    }

    @Override
    public int hashCode() {
        return Objects.hash(super.hashCode(), numberOfGears, wheelDiameter,
numberOfHelmets);
    }
}
```

- Schedule Class

```
package lk.oopCoursework1;

import java.util.Date;
import java.util.Objects;

public class Schedule {
    private Date pickUpDate;
    private Date dropOffDate;

    public Schedule(Date pickUpDate, Date dropOffDate) {
        this.pickUpDate = pickUpDate;
        this.dropOffDate = dropOffDate;
    }

    public Date getPickUpDate() {
        return pickUpDate;
    }

    public void setPickUpDate(Date pickUpDate) {
        this.pickUpDate = pickUpDate;
    }

    public Date getDropOffDate() {
        return dropOffDate;
    }

    public void setDropOffDate(Date dropOffDate) {
        this.dropOffDate = dropOffDate;
    }

    @Override
    public String toString() {
        return "Schedule{" +
                "pickUpDate=" + pickUpDate +
                ", dropOffDate=" + dropOffDate +
                '}';
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        Schedule schedule = (Schedule) o;
        return Objects.equals(pickUpDate, schedule.pickUpDate) &&
                Objects.equals(dropOffDate, schedule.dropOffDate);
    }

    @Override
    public int hashCode() {
        return Objects.hash(pickUpDate, dropOffDate);
    }
}
```

- WestminsterRentalVehicleManager Class

```
package lk.oopCoursework1;

import javax.swing.*;
import java.io.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.*;

public class WestminsterRentalManager implements RentalVehicleManager{

    private Car extendCar = new Car();
    private Motorbike extendmotorbike = new Motorbike();

    private  ArrayList<Vehicle> listOfVehicles;

    public WestminsterRentalManager(){
     listOfVehicles = new ArrayList<Vehicle>(); //creating the arraylist for cars
and motobikes.

     ConnectionClass connectionClass = new ConnectionClass(); //getting the database
connection
     Connection connection =connectionClass.getConnection();

     try{
         //getting the data of cars from the database and putting them in to the
array list.
         PreparedStatement selectCar =connection.prepareStatement("select * from
vehicles where(VehicleType='Car');");
         ResultSet resultSetCar = selectCar.executeQuery();
         while (resultSetCar.next()){
             Car objectCar = new Car(
                     resultSetCar.getString("VehiclePlateNumber"),
                     resultSetCar.getString("VehicleMake"),
                     resultSetCar.getInt("NumberOfPassengers"),
                     resultSetCar.getDouble("PricePerKM"),
                     resultSetCar.getString("VehicleType"),
                     resultSetCar.getInt("NumberOfAirbags"),
                     resultSetCar.getInt("FuelType"),
                     resultSetCar.getInt("NumberOfSeats")
             );
             listOfVehicles.add(objectCar);

             //deleting the cars from the database.
             Statement statement  =connection.createStatement();
             String sql = "delete from vehicles where VehicleType='Car'";
             statement.executeUpdate(sql);
         }
         //getting the data of motorbikes from the database and putting them in to
the array list.
         PreparedStatement selectBike = connection.prepareStatement("select * from
vehicles where (VehicleType='Motorbike');");
         ResultSet resultSetBike = selectBike.executeQuery();
         while (resultSetBike.next()){
             Motorbike objectBike = new Motorbike(
                     resultSetBike.getString("VehiclePlateNumber"),
                     resultSetBike.getString("VehicleMake"),
                     resultSetBike.getInt("NumberOfPassengers"),
                     resultSetBike.getDouble("PricePerKM"),
                     resultSetBike.getString("VehicleType"),
                     resultSetBike.getInt("NumberOfGears"),
                     resultSetBike.getInt("WheelDiameter"),
                     resultSetBike.getInt("NumberOfHelmets")
             );
             listOfVehicles.add(objectBike);

             //deleting the cars from the database.
             Statement statement  =connection.createStatement();
             String sql = "delete from vehicles where VehicleType='Motorbike'";
             statement.executeUpdate(sql);
         }
     }catch (Exception exc){
         exc.printStackTrace();
     }
    }

    public void addVehicle(Vehicle vehicle){
        if(listOfVehicles.size()<maxParking){
            listOfVehicles.add(vehicle);
        }
        else {
            System.out.println("Sorry the Parking is full!!");
        }
    }

    //method to add vehicles from the console to the array.
    @Override
    public void addVehicle() {
        //This will check whether the parking is full or not.
        if(listOfVehicles.size()<maxParking) {
            Scanner scannerOptionInput = new Scanner(System.in);
            System.out.print("\n" +
                    "Select vehicle type" +
                    "\n" +
                    "1. Car \n" +
                    "2. Motorbike \n" +
                    "3. Main Menu\n" +
                    "Choose: ");
            while (!scannerOptionInput.hasNextInt()) {
                System.out.println("Invalid Data Type!!");
                scannerOptionInput.next();
                System.out.print("Select vehicle type" +
                        "\n" +
                        "1. Car \n" +
                        "2. Motorbike \n" +
                        "3. Main Menu\n" +
                        "Choose: ");
            }
            int optionVehicle = scannerOptionInput.nextInt();

            while (optionVehicle != 3) {
                if (optionVehicle == 1) {

                    Scanner scannerCarInput = new Scanner(System.in);
                    System.out.print("Enter Plate number (WP ABC-1234): ");
                    String carPlateNumberInput =
scannerCarInput.nextLine().toUpperCase(); //plate number input.

                    System.out.print("Enter Make: ");
                    String carMakeInput = scannerCarInput.nextLine(); //make input.
                    System.out.print("Enter the number of Airbags: ");

                    while (!scannerCarInput.hasNextInt()) {
                        System.out.println("Invalid Data Type!!");
                        scannerCarInput.next();
                        System.out.print("Enter the number of Airbags: "); //number
of air bags input.
                    }
                    int carNumberOfAirbags = scannerCarInput.nextInt();

                    System.out.print("Enter the fuel fuel type (92, 95): ");
//getting the fuel type.
                    while (!scannerCarInput.hasNextInt()) {
                        System.out.println("Invalid Data Type!!");
                        scannerCarInput.next();
                        System.out.print("Enter the fuel fuel type (92, 95): ");
                    }
                    int carFuelTypeInput = scannerCarInput.nextInt();
                    while (!(carFuelTypeInput == 92 || carFuelTypeInput == 95)) {
                        System.out.println("Enter the correct fuel type!! \n" +
                                "Enter the fuel fuel type (92, 95):");
                        while (!scannerCarInput.hasNextInt()) {
                            System.out.println("Invalid Data Type!!");
                            scannerCarInput.next();
                            System.out.print("Enter the fuel fuel type (92, 95): ");
                        }
                        carFuelTypeInput = scannerCarInput.nextInt();
                    }

                    System.out.print("Enter the number of seats: "); //getting the
number of seats.
                    while (!scannerCarInput.hasNextInt()) {
                        System.out.println("Invalid Data Type!!");
                        scannerCarInput.next();
                        System.out.print("Enter the number of seats: ");
                    }

                    int carNumberOfSeats = scannerCarInput.nextInt();
                    int carNumberOfPassengers = 4;
                    double carPricePerKm = 50;
                    String carType = "Car";
                    extendCar.setNumberOfPassengers(carNumberOfPassengers);
//inserting the data to the arraylist.
                    extendCar.setPricePerKm(carPricePerKm);
                    extendCar.setVehicleType(carType);
                    Car car = new Car(carPlateNumberInput, carMakeInput,
carNumberOfPassengers, carPricePerKm, carType, carNumberOfAirbags, carFuelTypeInput,
carNumberOfSeats);
                    addVehicle(car);
                } else if (optionVehicle == 2) {

                    Scanner scannerBikeInput = new Scanner(System.in);//getting the
plate number.
                    System.out.print("Enter Plate number (WP ABC-1234): ");
                    String bikePlateNumberInput =
scannerBikeInput.nextLine().toUpperCase();

                    System.out.print("Enter Make: ");//getting the make.
                    String carMakeInput = scannerBikeInput.nextLine();

                    System.out.print("Enter the number of gears: ");//getting the
number of gears.
                    while (!scannerBikeInput.hasNextInt()) {
                        System.out.println("Invalid Data Type!!");
                        scannerBikeInput.next();
                        System.out.print("Enter the number of gears: ");
                    }
                    int bikeNumberOfGears = scannerBikeInput.nextInt();

                    System.out.print("Enter the wheel diameter (15, 16, 17):
");//getting the wheel diameter.
                    while (!scannerBikeInput.hasNextInt()) {
                        System.out.println("Invalid Data Type!!");
                        scannerBikeInput.next();
                        System.out.print("Enter the wheel diameter (15, 16, 17): ");
                    }
                    int bikeWheelDiameterInput = scannerBikeInput.nextInt();

                    System.out.print("Enter the number of helmets: ");//getting the
number of helmets.
                    while (!scannerBikeInput.hasNextInt()) {
                        System.out.println("Invalid Data Type!!");
                        scannerBikeInput.next();
                        System.out.print("Enter the number of helmets: ");
                    }
                    int bikeNumberOfHelmets = scannerBikeInput.nextInt();

                    int bikeNumberOfPassengers = 2;
                    double bikePricePerKm = 30;
                    String bikeType = "Motorbike";
                    extendmotorbike.setNumberOfPassengers(bikeNumberOfPassengers);
//inserting the data to the arraylist.
                    extendmotorbike.setPricePerKm(bikePricePerKm);
                    extendmotorbike.setVehicleType(bikeType);
                    Motorbike motorbike = new Motorbike(bikePlateNumberInput,
carMakeInput, bikeNumberOfPassengers, bikePricePerKm, bikeType, bikeNumberOfGears,
bikeWheelDiameterInput, bikeNumberOfHelmets);
                    addVehicle(motorbike);
                } else {
                    System.out.println("Invalid option!! Re-enter.. ");
                    break;
                }
                break;
            }
        }else{
            System.out.println("Sorry, the parking is full!!");
        }
    }

    //method to delete vehicles from the console to the array.
    @Override
    public void deleteVehicle() {
        //print the list of vehicles in the arraylist to make it easy to delete.
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("|%1$-20s|%2$-20s|%3$-20s|\n","       TYPE","    PLATE
NUMBER","        MAKE ");
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("%1$-20s%2$-20s%3$-20s\n","","","");
        for(int i=0; i<listOfVehicles.size(); i++){
            System.out.format("|%1$-20s|%2$-20s|%3$-
20s|\n",listOfVehicles.get(i).getVehicleType(),"
"+listOfVehicles.get(i).getPlateNumber(),"  "+listOfVehicles.get(i).getMake());
        }
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        Scanner deleteVehicleInput = new Scanner(System.in);
        System.out.print("\n" +
                "Select vehicle type you want to delete" +
                "\n" +
                "1. Car \n" +
                "2. Motorbike \n" +
                "3. Main Menu\n" +
                "Choose: ");
        while (!deleteVehicleInput.hasNextInt()){
            System.out.println("Invalid Data Type!!");
            deleteVehicleInput.next();
            System.out.print("Select vehicle type you want to delete" +
                    "\n" +
                    "1. Car \n" +
                    "2. Motorbike \n"+
                    "3. Main Menu\n" +
                    "Choose: ");
        }
        int optionDeleteVehicle = deleteVehicleInput.nextInt();
        while (optionDeleteVehicle!=3) {
            if (optionDeleteVehicle == 1) {
                String vehicleType = "Car";
                Scanner scannerDeleteVehicle = new Scanner(System.in);
                System.out.print("Enter the plate number of the vehicle that you
want to remove (WP ABC-1234): ");
```

```
                String deletePlateNumberInput =
scannerDeleteVehicle.nextLine().toUpperCase(); //getting the plate number to delete
the vehicle.
                if (listOfVehicles.removeIf(removeVehicle ->
removeVehicle.getPlateNumber().equals(deletePlateNumberInput))) { //deleting the
vehicle.
                    System.out.println("Plate Number " + deletePlateNumberInput + ",
" + vehicleType + " has been removed.");
                    System.out.println(50-listOfVehicles.size() + " spaces are
available for parking");
                } else {
                    System.out.println("Plate number does not exists!!");//if the
plate number is not available, it will pop this message.
                }
            } else if (optionDeleteVehicle == 2) {
                String vehicleType = "Motorbike";
                Scanner scannerDeleteVehicle = new Scanner(System.in);
                System.out.print("Enter the plate number of th vehicle that you want
to remove (WP ABC-1234): ");
                String deletePlateNumberInput =
scannerDeleteVehicle.nextLine().toUpperCase(); //getting the plate number to delete
the vehicle.
                if (listOfVehicles.removeIf(removeVehicle ->
removeVehicle.getPlateNumber().equals(deletePlateNumberInput))) { //deleting the
vehicle.
                    System.out.println("Plate Number " + deletePlateNumberInput + ",
" + vehicleType + " has been removed.");
                    System.out.println(50-listOfVehicles.size() + " spaces are
available for parking");
                } else {
                    System.out.println("Plate number does not exists!!"); //if the
plate number is not available, it will pop this message.
                }
            } else {
                System.out.println("Invalid option!! Re-enter.. ");
            }
            break;
        }
    }

    //printing the vehicle list.
    @Override
    public void printVehicleList() {
        Collections.sort(listOfVehicles, new Comparator<Vehicle>() { //sorting the
vehicle list according to the vehicle make.
            @Override
            public int compare(Vehicle o1, Vehicle o2) {
                return String.valueOf(o1.getMake()).compareTo(o2.getMake());
            }
            @Override
            public boolean equals(Object obj) {
                return false;
            }
        });
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("|%1$-20s|%2$-20s|%3$-20s|\n","       TYPE","    PLATE
NUMBER","        MAKE ");
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("%1$-20s%2$-20s%3$-20s\n","","","");
        for(int i=0; i<listOfVehicles.size(); i++){
            System.out.format("|%1$-20s|%2$-20s|%3$-
20s|\n",listOfVehicles.get(i).getVehicleType(),"
"+listOfVehicles.get(i).getPlateNumber(),"  "+listOfVehicles.get(i).getMake());
        }
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
    }

    //This method will save the arraylist to a text file.
    @Override
    public void saveVehicleList() {
        File file = new File("VehicleList.txt");
        try{
            FileWriter fileWriter = new FileWriter(file);
            Writer output = new BufferedWriter(fileWriter);
            int size = listOfVehicles.size();

            for(int i=0; i<size; i++ ){
                output.write(listOfVehicles.get(i).toString()+ "\n");
            }output.close();

        }catch (Exception e){
            JOptionPane.showMessageDialog(null,"Cannot create the file.");
        }
        System.out.println("Successfully save to the file VehicleList.txt");
    }

    //This method will put all the data in the arraylist to the sql data base for
later use.
    @Override
    public void saveInDatabase() {
        ConnectionClass connectionClass = new ConnectionClass(); //getting the
connection.
        Connection connection = connectionClass.getConnection();
        try {

            for (Vehicle vehicle:listOfVehicles){
                if(vehicle.getClass().equals(Car.class)){
                    PreparedStatement insertCar =connection.prepareStatement("insert
into vehicles (VehicleType, VehiclePlateNumber, PricePerKM, FuelType,
NumberOfPassengers," +
                            "NumberOfAirbags, NumberOfSeats, VehicleMake)" + "values
('Car',?,?,?,?,?,?,?)");
                    insertCar.setString(1,vehicle.getPlateNumber());
                    insertCar.setDouble(2,vehicle.getPricePerKm());
                    insertCar.setInt(3,((Car)vehicle).getFuelType());
                    insertCar.setInt(4,vehicle.getNumberOfPassengers());
                    insertCar.setInt(5,((Car)vehicle).getNumberOfAirbags());
                    insertCar.setInt(6,((Car)vehicle).getNumberOfSeats());
                    insertCar.setString(7,vehicle.getMake());
                    insertCar.execute();
                }
                else if(vehicle.getClass().equals(Motorbike.class)){
                    PreparedStatement insertMotorbike
=connection.prepareStatement("insert into vehicles (VehicleType, VehiclePlateNumber,
PricePerKM, NumberOfHelmets, NumberOfPassengers," +
                            "WheelDiameter, NumberOfGears, VehicleMake)" + "values
('Motorbike',?,?,?,?,?,?,?)");
                    insertMotorbike.setString(1,vehicle.getPlateNumber());
                    insertMotorbike.setDouble(2,vehicle.getPricePerKm());

insertMotorbike.setInt(3,((Motorbike)vehicle).getNumberOfHelmets());
                    insertMotorbike.setInt(4,vehicle.getNumberOfPassengers());

insertMotorbike.setInt(5,((Motorbike)vehicle).getWheelDiameter());

insertMotorbike.setInt(6,((Motorbike)vehicle).getNumberOfGears());
                    insertMotorbike.setString(7,vehicle.getMake());
                    insertMotorbike.execute();
                }
            }
        }catch (Exception exc){
            exc.printStackTrace();
        }

    }
}
```

- RentalVehicleManager

```
package lk.oopCoursework1;

public interface RentalVehicleManager {

    int maxParking = 50; //parking spaces
    void addVehicle();
    void deleteVehicle();
    void printVehicleList();
    void saveVehicleList();
    void saveInDatabase();
}
```

❖ Console menu functions

- Add a new vehicle

```
//method to add vehicles from the console to the array.
@Override
public void addVehicle() {
    //This will check whether the parking is full or not.
    if(listOfVehicles.size()<maxParking) {
        Scanner scannerOptionInput = new Scanner(System.in);
        System.out.print("\n" +
                "Select vehicle type" +
                "\n" +
                "1. Car \n" +
                "2. Motorbike \n" +
                "3. Main Menu\n" +
                "Choose: ");
        while (!scannerOptionInput.hasNextInt()) {
            System.out.println("Invalid Data Type!!");
            scannerOptionInput.next();
            System.out.print("Select vehicle type" +
                    "\n" +
                    "1. Car \n" +
                    "2. Motorbike \n" +
                    "3. Main Menu\n" +
                    "Choose: ");
        }
        int optionVehicle = scannerOptionInput.nextInt();

        while (optionVehicle != 3) {
            if (optionVehicle == 1) {

                Scanner scannerCarInput = new Scanner(System.in);
                System.out.print("Enter Plate number (WP ABC-1234): ");
                String carPlateNumberInput =
scannerCarInput.nextLine().toUpperCase(); //plate number input.

                System.out.print("Enter Make: ");
                String carMakeInput = scannerCarInput.nextLine(); //make input.
                System.out.print("Enter the number of Airbags: ");

                while (!scannerCarInput.hasNextInt()) {
                    System.out.println("Invalid Data Type!!");
                    scannerCarInput.next();
                    System.out.print("Enter the number of Airbags: "); //number of
air bags input.
                }
                int carNumberOfAirbags = scannerCarInput.nextInt();

                System.out.print("Enter the fuel fuel type (92, 95): "); //getting
the fuel type.
                while (!scannerCarInput.hasNextInt()) {
                    System.out.println("Invalid Data Type!!");
                    scannerCarInput.next();
                    System.out.print("Enter the fuel fuel type (92, 95): ");
                }
                int carFuelTypeInput = scannerCarInput.nextInt();
                while (!(carFuelTypeInput == 92 || carFuelTypeInput == 95)) {
                    System.out.println("Enter the correct fuel type!! \n" +
                            "Enter the fuel fuel type (92, 95):");
                    while (!scannerCarInput.hasNextInt()) {
                        System.out.println("Invalid Data Type!!");
                        scannerCarInput.next();
                        System.out.print("Enter the fuel fuel type (92, 95): ");
                    }
                    carFuelTypeInput = scannerCarInput.nextInt();
                }

                System.out.print("Enter the number of seats: "); //getting the
number of seats.
                while (!scannerCarInput.hasNextInt()) {
                    System.out.println("Invalid Data Type!!");
                    scannerCarInput.next();
                    System.out.print("Enter the number of seats: ");
                }

                int carNumberOfSeats = scannerCarInput.nextInt();
                int carNumberOfPassengers = 4;
                double carPricePerKm = 50;
                String carType = "Car";
                extendCar.setNumberOfPassengers(carNumberOfPassengers); //inserting
the data to the arraylist.
                extendCar.setPricePerKm(carPricePerKm);
                extendCar.setVehicleType(carType);
                Car car = new Car(carPlateNumberInput, carMakeInput,
carNumberOfPassengers, carPricePerKm, carType, carNumberOfAirbags, carFuelTypeInput,
carNumberOfSeats);
                addVehicle(car);
            } else if (optionVehicle == 2) {

                Scanner scannerBikeInput = new Scanner(System.in);//getting the
plate number.
                System.out.print("Enter Plate number (WP ABC-1234): ");
                String bikePlateNumberInput =
scannerBikeInput.nextLine().toUpperCase();

                System.out.print("Enter Make: ");//getting the make.
                String carMakeInput = scannerBikeInput.nextLine();

                System.out.print("Enter the number of gears: ");//getting the number
of gears.
                while (!scannerBikeInput.hasNextInt()) {
                    System.out.println("Invalid Data Type!!");
                    scannerBikeInput.next();
                    System.out.print("Enter the number of gears: ");
                }
                int bikeNumberOfGears = scannerBikeInput.nextInt();

                System.out.print("Enter the wheel diameter (15, 16, 17):
");//getting the wheel diameter.
                while (!scannerBikeInput.hasNextInt()) {
                    System.out.println("Invalid Data Type!!");
                    scannerBikeInput.next();
                    System.out.print("Enter the wheel diameter (15, 16, 17): ");
                }
                int bikeWheelDiameterInput = scannerBikeInput.nextInt();

                System.out.print("Enter the number of helmets: ");//getting the
number of helmets.
```

```
                while (!scannerBikeInput.hasNextInt()) {
                    System.out.println("Invalid Data Type!!");
                    scannerBikeInput.next();
                    System.out.print("Enter the number of helmets: ");
                }
                int bikeNumberOfHelmets = scannerBikeInput.nextInt();

                int bikeNumberOfPassengers = 2;
                double bikePricePerKm = 30;
                String bikeType = "Motorbike";
                extendmotorbike.setNumberOfPassengers(bikeNumberOfPassengers);
//inserting the data to the arraylist.
                extendmotorbike.setPricePerKm(bikePricePerKm);
                extendmotorbike.setVehicleType(bikeType);
                Motorbike motorbike = new Motorbike(bikePlateNumberInput,
carMakeInput, bikeNumberOfPassengers, bikePricePerKm, bikeType, bikeNumberOfGears,
bikeWheelDiameterInput, bikeNumberOfHelmets);
                addVehicle(motorbike);
            } else {
                System.out.println("Invalid option!! Re-enter.. ");
                break;
            }
            break;
        }
    }else{
        System.out.println("Sorry, the parking is full!!");
    }
}
```

- Delete vehicle

```
//method to delete vehicles from the console to the array.
@Override
public void deleteVehicle() {
    //print the list of vehicles in the arraylist to make it easy to delete.
    System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
    System.out.format("|%1$-20s|%2$-20s|%3$-20s|\n","       TYPE","    PLATE
NUMBER","        MAKE ");
    System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
    System.out.format("%1$-20s%2$-20s%3$-20s\n","","","");
    for(int i=0; i<listOfVehicles.size(); i++){
        System.out.format("|%1$-20s|%2$-20s|%3$-
20s|\n",listOfVehicles.get(i).getVehicleType(),"
"+listOfVehicles.get(i).getPlateNumber(),"  "+listOfVehicles.get(i).getMake());
    }
    System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
    Scanner deleteVehicleInput = new Scanner(System.in);
    System.out.print("\n" +
            "Select vehicle type you want to delete" +
            "\n" +
            "1. Car \n" +
            "2. Motorbike \n" +
            "3. Main Menu\n" +
            "Choose: ");
    while (!deleteVehicleInput.hasNextInt()){
        System.out.println("Invalid Data Type!!");
        deleteVehicleInput.next();
        System.out.print("Select vehicle type you want to delete" +
                "\n" +
                "1. Car \n" +
                "2. Motorbike \n"+
                "3. Main Menu\n" +
                "Choose: ");
    }
    int optionDeleteVehicle = deleteVehicleInput.nextInt();
    while (optionDeleteVehicle!=3) {
        if (optionDeleteVehicle == 1) {
            String vehicleType = "Car";
            Scanner scannerDeleteVehicle = new Scanner(System.in);
            System.out.print("Enter the plate number of the vehicle that you want to
remove (WP ABC-1234): ");
            String deletePlateNumberInput =
scannerDeleteVehicle.nextLine().toUpperCase(); //getting the plate number to delete
the vehicle.
            if (listOfVehicles.removeIf(removeVehicle ->
removeVehicle.getPlateNumber().equals(deletePlateNumberInput))) { //deleting the
vehicle.
                System.out.println("Plate Number " + deletePlateNumberInput + ", " +
vehicleType + " has been removed.");
                System.out.println(50-listOfVehicles.size() + " spaces are available
for parking");
            } else {
                System.out.println("Plate number does not exists!!");//if the plate
number is not available, it will pop this message.
            }
        } else if (optionDeleteVehicle == 2) {
            String vehicleType = "Motorbike";
            Scanner scannerDeleteVehicle = new Scanner(System.in);
            System.out.print("Enter the plate number of th vehicle that you want to
remove (WP ABC-1234): ");
            String deletePlateNumberInput =
scannerDeleteVehicle.nextLine().toUpperCase(); //getting the plate number to delete
the vehicle.
            if (listOfVehicles.removeIf(removeVehicle ->
removeVehicle.getPlateNumber().equals(deletePlateNumberInput))) { //deleting the
vehicle.
                System.out.println("Plate Number " + deletePlateNumberInput + ", " +
vehicleType + " has been removed.");
                System.out.println(50-listOfVehicles.size() + " spaces are available
for parking");
            } else {
                System.out.println("Plate number does not exists!!"); //if the plate
number is not available, it will pop this message.
            }
        } else {
            System.out.println("Invalid option!! Re-enter.. ");
        }
        break;
    }
}
```

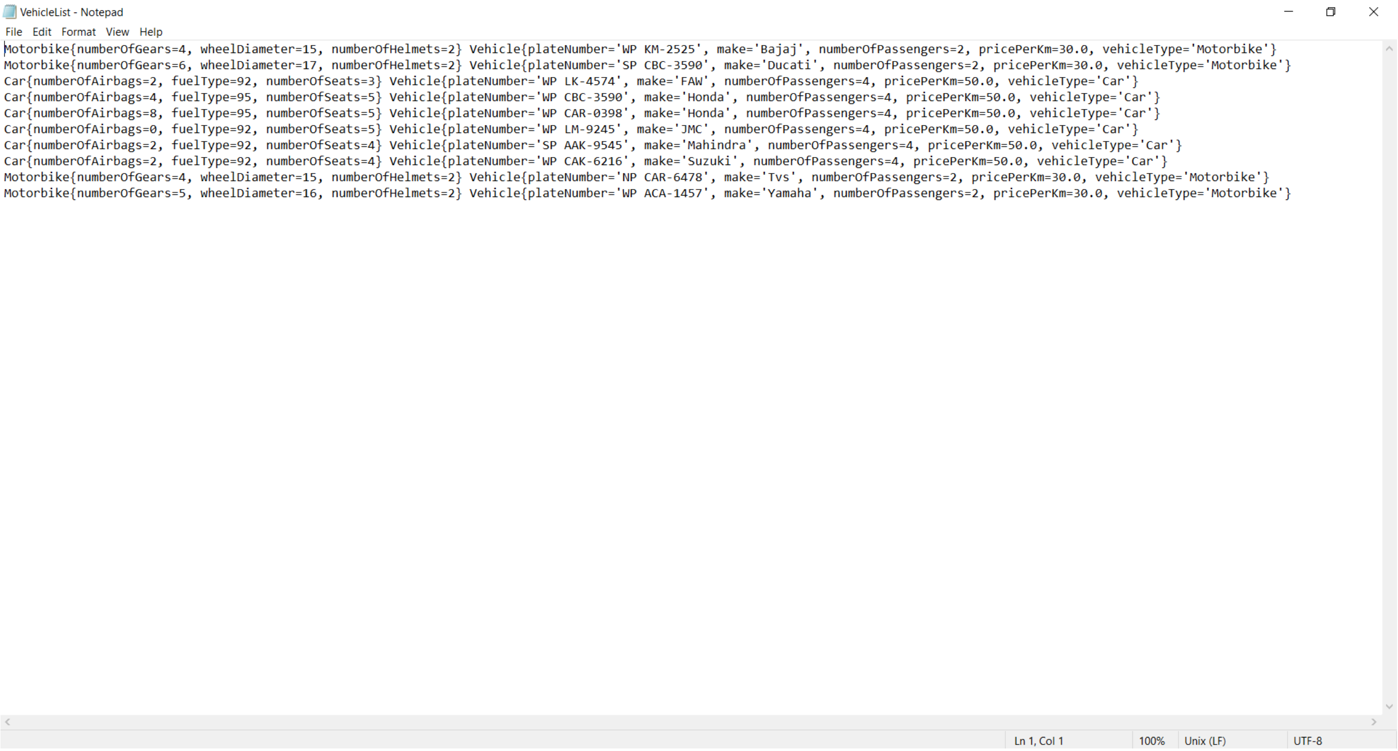
- Print the list of the vehicles and sorting

```
//printing the vehicle list.
@Override
public void printVehicleList() {
    Collections.sort(listOfVehicles, new Comparator<Vehicle>() { //sorting the 
vehicle list according to the vehicle make.
        @Override
        public int compare(Vehicle o1, Vehicle o2) {
            return String.valueOf(o1.getMake()).compareTo(o2.getMake());
        }
        @Override
        public boolean equals(Object obj) {
            return false;
        }
    });
    System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
    System.out.format("|%1$-20s|%2$-20s|%3$-20s|\n","       TYPE","    PLATE 
NUMBER","        MAKE ");
    System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
    System.out.format("%1$-20s%2$-20s%3$-20s\n","","","");
    for(int i=0; i<listOfVehicles.size(); i++){
        System.out.format("|%1$-20s|%2$-20s|%3$-
20s|\n",listOfVehicles.get(i).getVehicleType()," 
"+listOfVehicles.get(i).getPlateNumber(),"  "+listOfVehicles.get(i).getMake());
    }
    System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
}
```

- Write/ save the vehicle list

```
//This method will save the arraylist to a text file.
@Override
public void saveVehicleList() {
    File file = new File("VehicleList.txt");
    try{
        FileWriter fileWriter = new FileWriter(file);
        Writer output = new BufferedWriter(fileWriter);
        int size = listOfVehicles.size();

        for(int i=0; i<size; i++ ){
            output.write(listOfVehicles.get(i).toString()+ "\n");
        }output.close();

    }catch (Exception e){
        JOptionPane.showMessageDialog(null,"Cannot create the file.");
    }
    System.out.println("Successfully save to the file VehicleList.txt");
}
```

- Snippet of the notepad file



```
Motorbike{numberOfGears=4, wheelDiameter=15, numberOfHelmets=2} Vehicle{plateNumber='WP KM-2525', make='Bajaj', numberOfPassengers=2, pricePerKm=30.0, vehicleType='Motorbike'}
Motorbike{numberOfGears=6, wheelDiameter=17, numberOfHelmets=2} Vehicle{plateNumber='SP CBC-3590', make='Ducati', numberOfPassengers=2, pricePerKm=30.0, vehicleType='Motorbike'}
Car{numberOfAirbags=2, fuelType=92, numberOfSeats=3} Vehicle{plateNumber='WP LK-4574', make='FAW', numberOfPassengers=4, pricePerKm=50.0, vehicleType='Car'}
Car{numberOfAirbags=4, fuelType=95, numberOfSeats=5} Vehicle{plateNumber='WP CBC-3590', make='Honda', numberOfPassengers=4, pricePerKm=50.0, vehicleType='Car'}
Car{numberOfAirbags=8, fuelType=95, numberOfSeats=5} Vehicle{plateNumber='WP CAR-0398', make='Honda', numberOfPassengers=4, pricePerKm=50.0, vehicleType='Car'}
Car{numberOfAirbags=0, fuelType=92, numberOfSeats=5} Vehicle{plateNumber='WP LM-9245', make='JMC', numberOfPassengers=4, pricePerKm=50.0, vehicleType='Car'}
Car{numberOfAirbags=2, fuelType=92, numberOfSeats=4} Vehicle{plateNumber='SP AAK-9545', make='Mahindra', numberOfPassengers=4, pricePerKm=50.0, vehicleType='Car'}
Car{numberOfAirbags=2, fuelType=92, numberOfSeats=4} Vehicle{plateNumber='WP CAK-6216', make='Suzuki', numberOfPassengers=4, pricePerKm=50.0, vehicleType='Car'}
Motorbike{numberOfGears=4, wheelDiameter=15, numberOfHelmets=2} Vehicle{plateNumber='NP CAR-6478', make='Tvs', numberOfPassengers=2, pricePerKm=30.0, vehicleType='Motorbike'}
Motorbike{numberOfGears=5, wheelDiameter=16, numberOfHelmets=2} Vehicle{plateNumber='WP ACA-1457', make='Yamaha', numberOfPassengers=2, pricePerKm=30.0, vehicleType='Motorbike'}
```

❖ Graphical User Interface

- User class to retrieve the data from the data base.

```
package lk.oopCoursework1;

import javafx.beans.property.SimpleDoubleProperty;
import javafx.beans.property.SimpleIntegerProperty;
import javafx.beans.property.SimpleStringProperty;

public class User {
    private final SimpleStringProperty vehicleType;
    private final SimpleStringProperty vehiclePlateNumber;
    private final SimpleDoubleProperty pricePerKm;
    private final SimpleIntegerProperty fuelType;
    private final SimpleIntegerProperty numberOfHelmets;
    private final SimpleIntegerProperty numberOfPassengers;
    private final SimpleIntegerProperty numberOfAirbags;
    private final SimpleIntegerProperty numberOfSeats;
    private final SimpleIntegerProperty numberOfGears;
    private final SimpleIntegerProperty wheelDiameter;
    private final SimpleStringProperty vehicleMake;

    public User(String type, String plateNumber, double price, int fuel, int
helmets, int passengers, int airbags, int seats, int gears, int diameter, String
make) {
        this.vehicleType = new SimpleStringProperty(type);
        this.vehiclePlateNumber = new SimpleStringProperty(plateNumber);
        this.pricePerKm = new SimpleDoubleProperty(price);
        this.fuelType = new SimpleIntegerProperty(fuel);
        this.numberOfHelmets = new SimpleIntegerProperty(helmets);
        this.numberOfPassengers = new SimpleIntegerProperty(passengers);
        this.numberOfAirbags = new SimpleIntegerProperty(airbags);
        this.numberOfSeats = new SimpleIntegerProperty(seats);
        this.numberOfGears = new SimpleIntegerProperty(gears);
        this.wheelDiameter = new SimpleIntegerProperty(diameter);
        this.vehicleMake = new SimpleStringProperty(make);
    }

    public String getVehicleType(){
        return vehicleType.get();
    }

    public String getVehiclePlateNumber(){
        return vehiclePlateNumber.get();
    }

    public double getPricePerKM(){
        return pricePerKm.get();
    }

    public int getFuelType() {
        return fuelType.get();
    }

    public int getNumberOfHelmets() {
        return numberOfHelmets.get();
    }

    public int getNumberOfPassengers() {
```

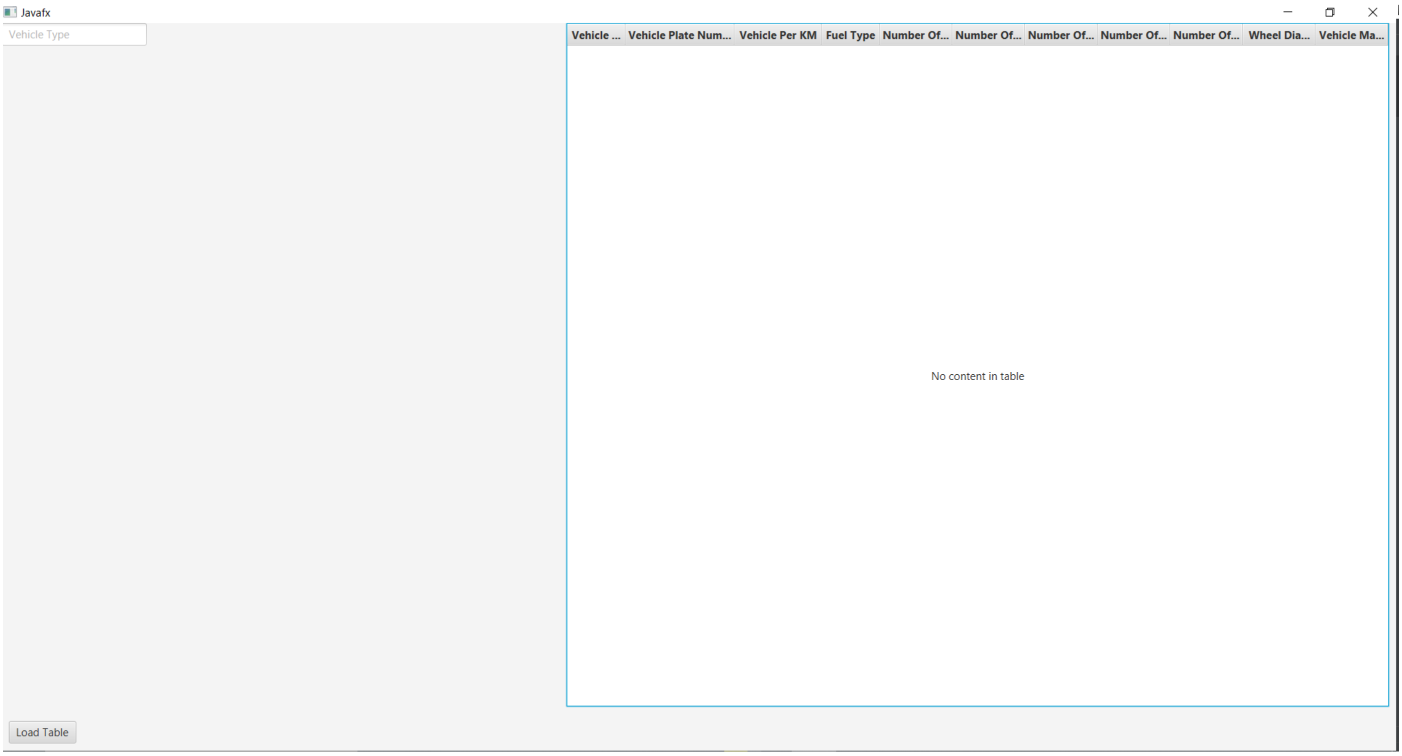
```
    return numberOfPassengers.get();
}

public int getNumberOfAirbags() {
    return numberOfAirbags.get();
}

public int getNumberOfSeats() {
    return numberOfSeats.get();
}

public int getNumberOfGears() {
    return numberOfGears.get();
}

public int getWheelDiameter() {
    return wheelDiameter.get();
}

public String getVehicleMake() {
    return vehicleMake.get();
}

public void setVehicleType(String type){
    vehicleType.set(type);
}

public void setVehiclePlateNumber(String plateNumber){
    vehiclePlateNumber.set(plateNumber);
}

public void setPricePerKm(double price){
    pricePerKm.set(price);
}

public void setFuelType(int fuel){
    fuelType.set(fuel);
}

public void setNumberOfHelmets(int helmets){
    numberOfHelmets.set(helmets);
}

public void setNumberOfPassengers(int passengers){
    numberOfPassengers.set(passengers);
}

public void setNumberOfAirbags(int airbags){
    numberOfPassengers.set(airbags);
}

public void setNumberOfSeats(int seats){
    numberOfSeats.set(seats);
}

public void setNumberOfGears(int gears){
    numberOfGears.set(gears);
}
```

```
    public void setWheelDiameter(int diameter){
        wheelDiameter.set(diameter);
    }

    public void setVehicleMake(String make){
        vehicleMake.set(make);
    }
}
```

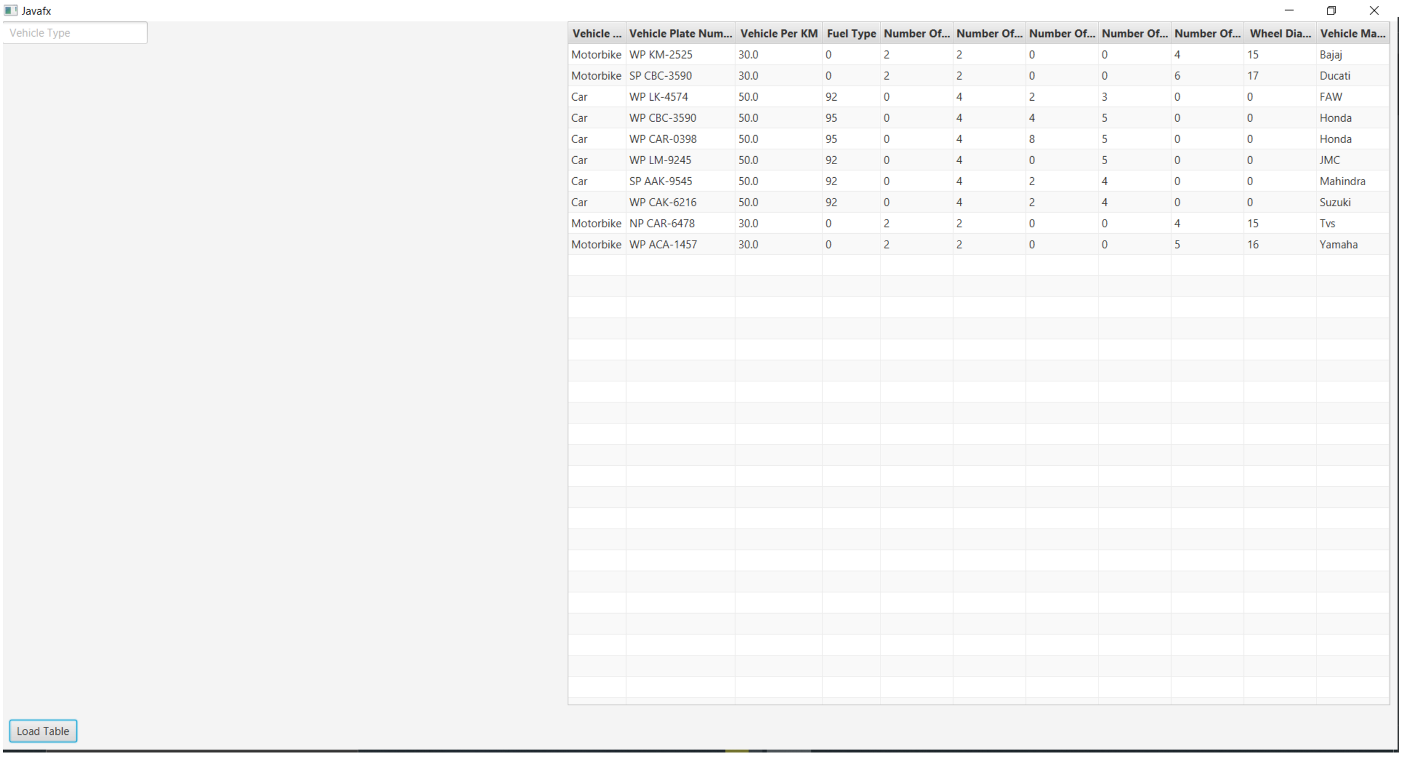
- Gui class which display the vehicle list and let the user to filter it according to the vehicle type

```
package lk.oopCoursework1;
import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.collections.transformation.FilteredList;
import javafx.collections.transformation.SortedList;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.layout.BorderPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.paint.Color;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.function.Predicate;

public class Gui extends Application {
    Connection conn;
    PreparedStatement preparedStatement = null;
    ResultSet resultSet = null;
    TextField searchField;


    public static void main(String[] args) {
        launch(args);
    }

    public void CheckConnection(){
        conn = SqlConnection.DbConnector();
        if(conn == null){
            System.out.println("Connection is not successful.");
            System.exit(1);
        }else {
            System.out.println("Connection is successful.");
        }
    }

    @Override
    public void start(Stage primaryStage) throws Exception {
        CheckConnection();

        primaryStage.setTitle("Javafx");
        BorderPane layout = new BorderPane();
        Scene newScene = new Scene(layout,1500,1000,Color.rgb(0,0,0,0));

        TableView<User> table = new TableView<>();
        final ObservableList<User> data = FXCollections.observableArrayList();

        TableColumn column1 = new TableColumn("Vehicle Type");
        column1.setMinWidth(50);
        column1.setCellValueFactory(new PropertyValueFactory<>("vehicleType"));

        TableColumn column2 = new TableColumn("Vehicle Plate Number");
        column2.setMinWidth(150);
        column2.setCellValueFactory(new
PropertyValueFactory<>("vehiclePlateNumber"));

        TableColumn column3 = new TableColumn("Vehicle Per KM");
        column3.setMinWidth(120);
        column3.setCellValueFactory(new PropertyValueFactory<>("pricePerKM"));

        TableColumn column4 = new TableColumn("Fuel Type");
        column4.setMinWidth(80);
        column4.setCellValueFactory(new PropertyValueFactory<>("fuelType"));

        TableColumn column5 = new TableColumn("Number Of Helmets");
        column5.setMinWidth(100);
        column5.setCellValueFactory(new PropertyValueFactory<>("numberOfHelmets"));

        TableColumn column6 = new TableColumn("Number Of Passengers");
        column6.setMinWidth(100);
        column6.setCellValueFactory(new
PropertyValueFactory<>("numberOfPassengers"));

        TableColumn column7 = new TableColumn("Number Of Airbags");
        column7.setMinWidth(100);
        column7.setCellValueFactory(new PropertyValueFactory<>("numberOfAirbags"));

        TableColumn column8 = new TableColumn("Number Of Seats");
        column8.setMinWidth(100);
        column8.setCellValueFactory(new PropertyValueFactory<>("numberOfSeats"));

        TableColumn column9 = new TableColumn("Number Of Gears");
        column9.setMinWidth(100);
        column9.setCellValueFactory(new PropertyValueFactory<>("numberOfGears"));

        TableColumn column10 = new TableColumn("Wheel Diameter");
        column10.setMinWidth(100);
        column10.setCellValueFactory(new PropertyValueFactory<>("wheelDiameter"));

        TableColumn column11 = new TableColumn("Vehicle Make");
        column11.setMinWidth(100);
        column11.setCellValueFactory(new PropertyValueFactory<>("vehicleMake"));


table.getColumns().addAll(column1,column2,column3,column4,column5,column6,column7,co
lumn8,column9,column10,column11);
        layout.setRight(table);
        BorderPane.setMargin(table,new Insets(0,10,10,0));

        Button load = new Button("Load Table");
        load.setFont(Font.font("SanSerif",15));
        load.setOnAction(e->{
            try{
                String query = "select * from vehicles";
                preparedStatement = conn.prepareStatement(query);
                resultSet = preparedStatement.executeQuery();

                while (resultSet.next()){
                    data.add(new User(
                            resultSet.getString("VehicleType"),
                            resultSet.getString("VehiclePlateNumber"),
                            resultSet.getDouble("PricePerKM"),
                            resultSet.getInt("FuelType"),
                            resultSet.getInt("NumberOfHelmets"),
                            resultSet.getInt("NumberOfPassengers"),
                            resultSet.getInt("NumberOfAirbags"),
                            resultSet.getInt("NumberOfSeats"),
                            resultSet.getInt("NumberOfGears"),
                            resultSet.getInt("WheelDiameter"),
                            resultSet.getString("VehicleMake")
                    ));
                    table.setItems(data);

                }
                preparedStatement.close();
                resultSet.close();
            }catch (Exception e2){
                System.err.println(e2);
            }
        });

        HBox hBox = new HBox(5);
        hBox.getChildren().add(load);
        layout.setBottom(hBox);
        BorderPane.setMargin(hBox, new Insets(10,0,10,10));

        VBox fields = new VBox(5);
        searchField = new TextField();
        searchField.setFont(Font.font("SanSerif",15));
        searchField.setPromptText("Vehicle Type");
        searchField.setMaxWidth(200);

        fields.getChildren().addAll(searchField);
        layout.setCenter(fields);
        FilteredList<User> filteredList = new FilteredList<>(data, e-> true);
        searchField.setOnKeyReleased(e->{
            searchField.textProperty().addListener((observable, oldValue, newValue)
->{
                filteredList.setPredicate((Predicate <? super  User>) user->{
                    if (newValue == null || newValue.isEmpty()){
                        return true;
                    }
                    String lowerCaseFilter = newValue;
                    if (user.getVehicleType().contains(newValue)){
                        return true;
                    }else if(user.getVehicleType().contains(lowerCaseFilter)){
                        return true;
                    }
                    return false;
                });
            });

            SortedList<User> sortedData = new SortedList<>(filteredList);
            sortedData.comparatorProperty().bind(table.comparatorProperty());
            table.setItems(sortedData);
        });
        primaryStage.setScene(newScene);
        primaryStage.show();

    }
}
```

- Gui Snippets
- Graphical User Interface

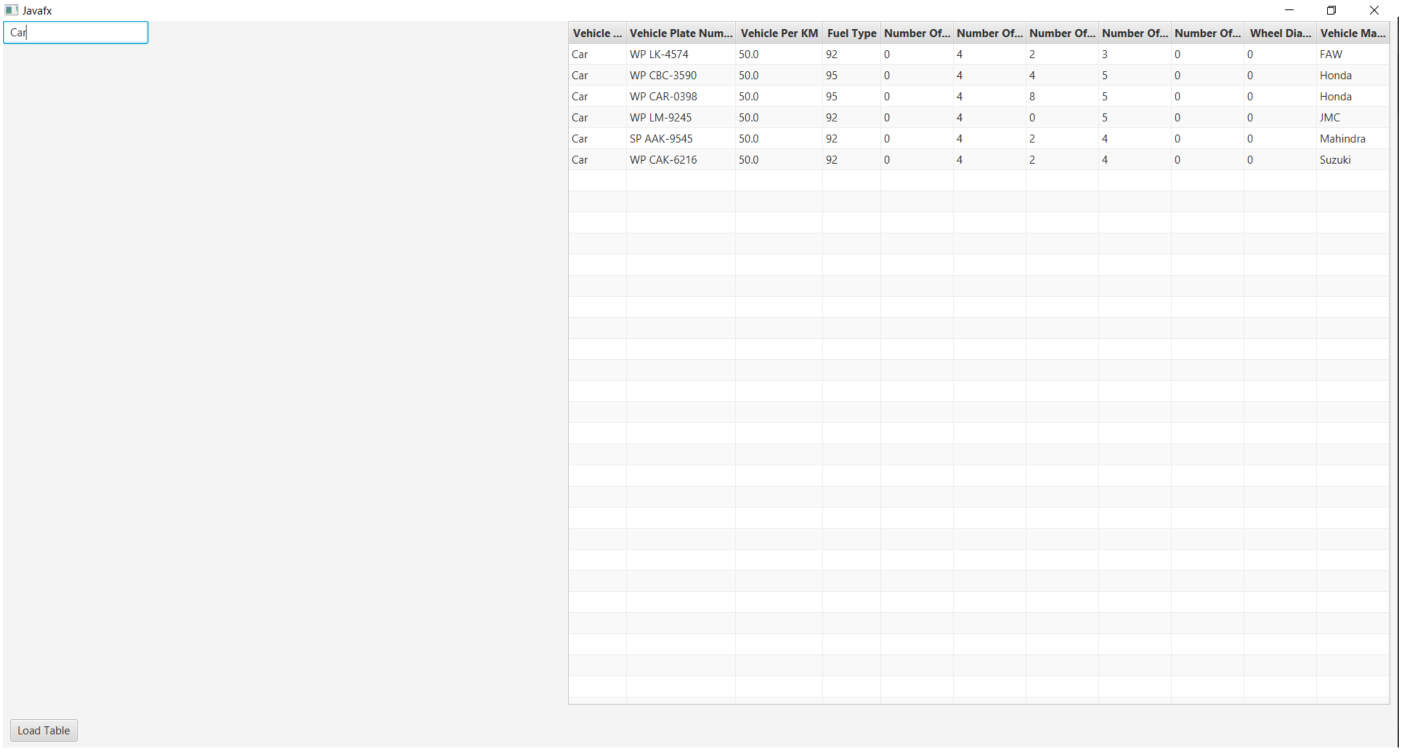


- Graphical User Interface with Data



Vehicle ...	Vehicle Plate Num...	Vehicle Per KM	Fuel Type	Number Of...	Number Of...	Number Of...	Number Of...	Number Of...	Wheel Dia...	Vehicle Ma...
Motorbike	WP KM-2525	30.0	0	2	2	0	0	4	15	Bajaj
Motorbike	SP CBC-3590	30.0	0	2	2	0	0	6	17	Ducati
Car	WP LK-4574	50.0	92	0	4	2	3	0	0	FAW
Car	WP CBC-3590	50.0	95	0	4	4	5	0	0	Honda
Car	WP CAR-0398	50.0	95	0	4	8	5	0	0	Honda
Car	WP LM-9245	50.0	92	0	4	0	5	0	0	JMC
Car	SP AAK-9545	50.0	92	0	4	2	4	0	0	Mahindra
Car	WP CAK-6216	50.0	92	0	4	2	4	0	0	Suzuki
Motorbike	NP CAR-6478	30.0	0	2	2	0	0	4	15	Tvs
Motorbike	WP ACA-1457	30.0	0	2	2	0	0	5	16	Yamaha

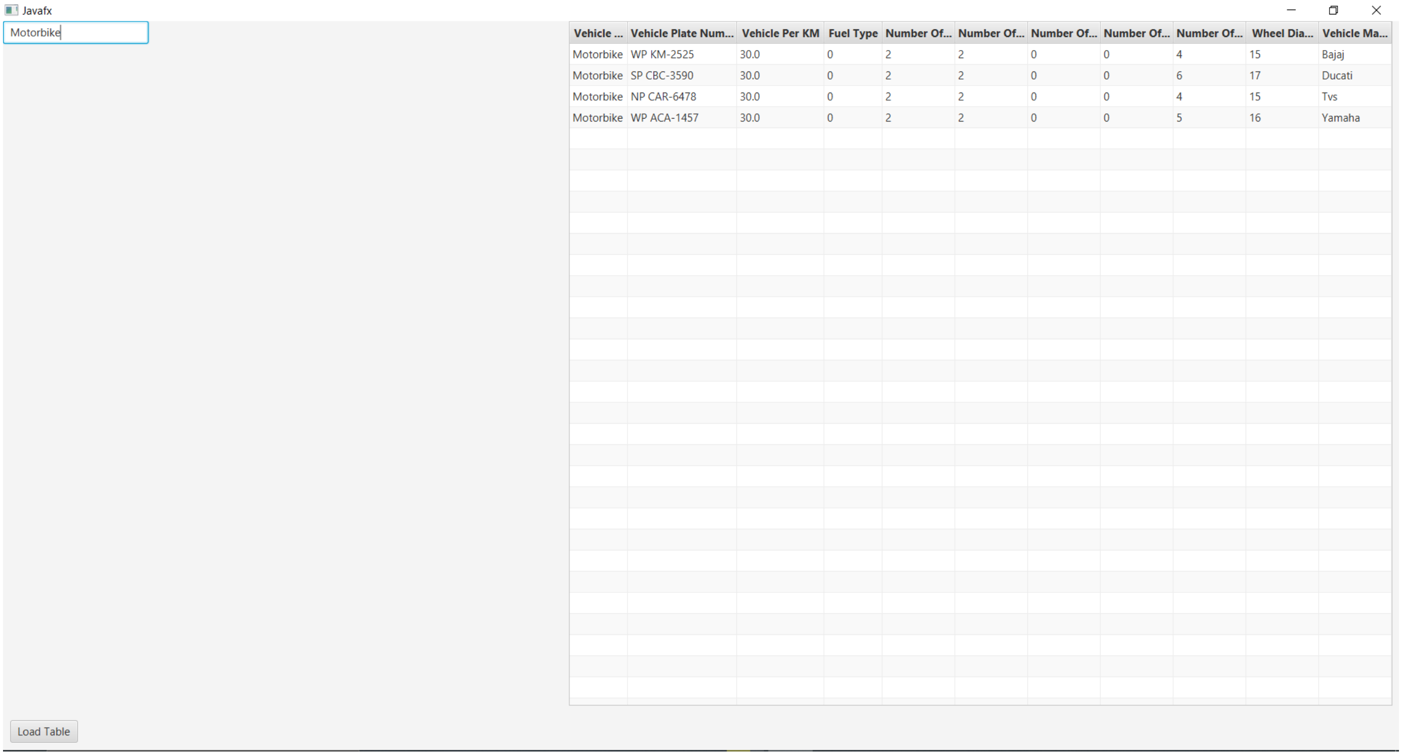
- Filtered according to the Vehicle Type – Car



Car

Vehicle ...	Vehicle Plate Num...	Vehicle Per KM	Fuel Type	Number Of...	Number Of...	Number Of...	Number Of...	Number Of...	Wheel Dia...	Vehicle Ma...
Car	WP LK-4574	50.0	92	0	4	2	3	0	0	FAW
Car	WP CBC-3590	50.0	95	0	4	4	5	0	0	Honda
Car	WP CAR-0398	50.0	95	0	4	8	5	0	0	Honda
Car	WP LM-9245	50.0	92	0	4	0	5	0	0	JMC
Car	SP AAK-9545	50.0	92	0	4	2	4	0	0	Mahindra
Car	WP CAK-6216	50.0	92	0	4	2	4	0	0	Suzuki

- Filtered according to the Vehicle Type -Motorbike



Vehicle ...	Vehicle Plate Num...	Vehicle Per KM	Fuel Type	Number Of...	Number Of...	Number Of...	Number Of...	Number Of...	Wheel Dia...	Vehicle Ma...
Motorbike	WP KM-2525	30.0	0	2	2	0	0	4	15	Bajaj
Motorbike	SP CBC-3590	30.0	0	2	2	0	0	6	17	Ducati
Motorbike	NP CAR-6478	30.0	0	2	2	0	0	4	15	Tvs
Motorbike	WP ACA-1457	30.0	0	2	2	0	0	5	16	Yamaha

❖ Testing and system validation

• Test plan

Test Plan Design

No	Test Case	Input	Expected Output	Actual Output	Errors
1	Main menu	1	Select vehicle type	Select vehicle type	No
			1. Car	1. Car	No
			2. Motorbike	2. Motorbike	No
			3. Main Menu	3. Main Menu	No
			Choose:	Choose:	No
		2	Select vehicle type you want to delete	Select vehicle type you want to delete	No
			1. Car	1. Car	No
			2. Motorbike	2. Motorbike	No
			3. Main Menu	3. Main Menu	No
			Choose:	Choose:	No
		3	Print the table	Print the table	No
		4	Successfully save to the file VehicleList.txt	Successfully save to the file VehicleList.txt	No
		5	Open Gui	Open Gui	No
		6	------->> Programme End <<--------	------->> Programme End <<--------	No
		-1	Please the choose the correct option!!	Please the choose the correct option!!	No
		0	Please the choose the correct option!!	Please the choose the correct option!!	No
		20	Please the choose the correct option!!	Please the choose the correct option!!	No
		@#!	'@#!' is an invalid data type!!	'@#!' is an invalid data type!!	No
		a	'a' is an invalid data type!!	'a' is an invalid data type!!	No
2	Add car	1	Enter Plate number (WP ABC-1234):	Enter Plate number (WP ABC-1234):	No
		2	Enter Plate number (WP ABC-1234):	Enter Plate number (WP ABC-1234):	No
		3	Go back to the main menu	Go back to the main menu	No
		0	Invalid option!! Re-enter..	Invalid option!! Re-enter..	No
		-10	Invalid option!! Re-enter..	Invalid option!! Re-enter..	No
		fsdg	'fsdg' is an invalid data type!!	'fsdg' is an invalid data type!!	No
		@!!#	Invalid Data Type!!	Invalid Data Type!!	No
3	Enter plate	WP ABC-1234	Accept Any String	Accept Any String	No
4	Enter Make	Toyota	Accept Any String	Accept Any String	No
5	No. Of Airbags	2	Accept and move forward	Accept and move forward	No
		0	Accept and move forward	Accept and move forward	No
		jgf	Invalid Data Type!!	Invalid Data Type!!	No
6	Fuelt Type	92	Accept and move forward	Accept and move forward	No
		95	Accept and move forward	Accept and move forward	No
		20	Enter the correct fuel type!!	Enter the correct fuel type!!	No
		-1	Enter the correct fuel type!!	Enter the correct fuel type!!	No
		sdf	Invalid Data Type!!	Invalid Data Type!!	No
7	No. Of Seats	2	Accept and move forward	Accept and move forward	No
		sgasg	Invalid Data Type!!	Invalid Data Type!!	No
8	No. Of Gears	5	Accept and move forward	Accept and move forward	No
		3	Accept and move forward	Accept and move forward	No
		sdf	Invalid Data Type!!	Invalid Data Type!!	No
9	Wheel Diameter	16	Accept and move forward	Accept and move forward	No
		17	Accept and move forward	Accept and move forward	No

		dfg	Invalid Data Type!!	Invalid Data Type!!	No
10	No. Of Helmets	2	Accept and move forward	Accept and move forward	No
		4	Accept and move forward	Accept and move forward	No
		sfsdf	Invalid Data Type!!	Invalid Data Type!!	No
11	Delete Vehicle	1	Ask for the plate number	Ask for the plate number	No
		2	Ask for the plate number	Ask for the plate number	No
		3	Main Menu	Main Menu	No

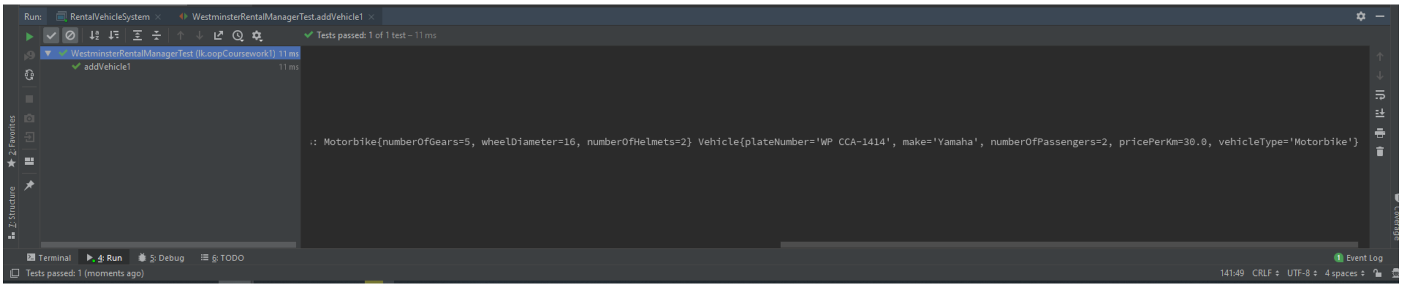
- Automated testing / Unit testing

- Testing class code

```
package lk.oopCoursework1;

import org.junit.Test;

import javax.swing.*;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.Writer;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.Scanner;

import static org.junit.Assert.*;

public class WestminsterRentalManagerTest implements RentalVehicleManager {

    private Car extendCar = new Car();
    private Motorbike extendmotorbike = new Motorbike();

    private ArrayList<Vehicle> listOfVehicles;
    private Vehicle vehicle;

    public WestminsterRentalManagerTest() {
        listOfVehicles = new ArrayList<Vehicle>();
    }

    @Test
    public void addVehicle() {
        if(listOfVehicles.size()<maxParking){
            listOfVehicles.add(vehicle);
        }
        else {
            System.out.println("Sorry the Parking is full!!");
        }
    }

    @Test
    public void addVehicle1() {
        if(listOfVehicles.size()<maxParking) {
            Scanner scannerOptionInput = new Scanner(System.in);
            System.out.print("\n" +
                    "Select vehicle type" +
                    "\n" +
                    "1. Car \n" +
                    "2. Motorbike \n" +
                    "3. Main Menu\n" +
                    "Choose: ");
            int optionVehicle = 1;
            while (!(optionVehicle==(int)optionVehicle)) {
                System.out.println("Invalid Data Type!!");
//                scannerOptionInput.next();
                System.out.print("Select vehicle type" +
                        "\n" +
                        "1. Car \n" +
                        "2. Motorbike \n" +
                        "3. Main Menu\n" +
                        "Choose: ");
            }
            optionVehicle = 2;

            while (optionVehicle != 3) {
                if (optionVehicle == 1) {

                    Scanner scannerCarInput = new Scanner(System.in);
                    System.out.print("Enter Plate number (WP ABC-1234): ");
                    String carPlateNumberInput = "WP ABC-1234"; //plate number
input.

                    System.out.print("Enter Make: ");
                    String carMakeInput = "Toyota"; //make input.

                    int carNumberOfAirbags = 2;
                    System.out.print("Enter the number of Airbags: ");
                    while (!(carNumberOfAirbags==(int)carNumberOfAirbags)) {
                        System.out.println("Invalid Data Type!!");
                        scannerCarInput.next();
                        System.out.print("Enter the number of Airbags: "); //number
of air bags input.
                    }
                    carNumberOfAirbags  = 2;


                    int carFuelTypeInput =92;
                    System.out.print("Enter the fuel fuel type (92, 95): ");
//getting the fuel type.
                    while (!(carFuelTypeInput==(int)carFuelTypeInput)) {
                        System.out.println("Invalid Data Type!!");
//                        scannerCarInput.next();
                        System.out.print("Enter the fuel fuel type (92, 95): ");
                    }
                    carFuelTypeInput = 92;
                    while (!(carFuelTypeInput == 92 || carFuelTypeInput == 95)) {
                        System.out.println("Enter the correct fuel type!! \n" +
                                "Enter the fuel fuel type (92, 95):");
                        while (!(carFuelTypeInput==(int)carFuelTypeInput)) {
                            System.out.println("Invalid Data Type!!");
//                            scannerCarInput.next();
                            System.out.print("Enter the fuel fuel type (92, 95): ");
                        }
                        carFuelTypeInput = 92;
                    }
                    int carNumberOfSeats = 4;
                    System.out.print("Enter the number of seats: "); //getting the 
number of seats.
                    while (!(carNumberOfSeats==(int)carNumberOfSeats)) {
                        System.out.println("Invalid Data Type!!");
//                        scannerCarInput.next();
                        System.out.print("Enter the number of seats: ");
                    }

                    carNumberOfSeats = 4;

                    int carNumberOfPassengers = 4;
                    double carPricePerKm = 50;
                    String carType = "Car";
                    extendCar.setNumberOfPassengers(carNumberOfPassengers);
//inserting the data to the arraylist.
                    extendCar.setPricePerKm(carPricePerKm);
                    extendCar.setVehicleType(carType);
                    Car car = new Car(carPlateNumberInput, carMakeInput, 
carNumberOfPassengers, carPricePerKm, carType, carNumberOfAirbags, carFuelTypeInput, 
carNumberOfSeats);
                    addVehicle(car);
                    System.out.println(car);
                } else if (optionVehicle == 2) {

                    Scanner scannerBikeInput = new Scanner(System.in);//getting the 
plate number.
                    System.out.print("Enter Plate number (WP ABC-1234): ");
                    String bikePlateNumberInput = "WP CCA-1414";

                    System.out.print("Enter Make: ");//getting the make.
                    String carMakeInput = "Yamaha";

                    int bikeNumberOfGears = 5;
                    System.out.print("Enter the number of gears: ");//getting the 
number of gears.
                    while (!(bikeNumberOfGears==(int)bikeNumberOfGears)) {
                        System.out.println("Invalid Data Type!!");
//                        scannerBikeInput.next();
                        System.out.print("Enter the number of gears: ");
                    }
                    bikeNumberOfGears = 5;

                    int bikeWheelDiameterInput = 16;
                    System.out.print("Enter the wheel diameter (15, 16, 17): 
");//getting the wheel diameter.
                    while (!(bikeNumberOfGears==(int)bikeNumberOfGears)) {
                        System.out.println("Invalid Data Type!!");
//                        scannerBikeInput.next();
                        System.out.print("Enter the wheel diameter (15, 16, 17): ");
                    }
                    bikeWheelDiameterInput = 16;

                    int bikeNumberOfHelmets = 2;

                    System.out.print("Enter the number of helmets: ");//getting the 
number of helmets.
                    while (!(bikeNumberOfHelmets==(int)bikeNumberOfHelmets)) {
                        System.out.println("Invalid Data Type!!");
//                        scannerBikeInput.next();
                        System.out.print("Enter the number of helmets: ");
                    }
                    bikeNumberOfHelmets = 2;

                    int bikeNumberOfPassengers = 2;
                    double bikePricePerKm = 30;
                    String bikeType = "Motorbike";
                    extendmotorbike.setNumberOfPassengers(bikeNumberOfPassengers);
//inserting the data to the arraylist.
                    extendmotorbike.setPricePerKm(bikePricePerKm);
                    extendmotorbike.setVehicleType(bikeType);
                    Motorbike motorbike = new Motorbike(bikePlateNumberInput,
carMakeInput, bikeNumberOfPassengers, bikePricePerKm, bikeType, bikeNumberOfGears,
bikeWheelDiameterInput, bikeNumberOfHelmets);
                    addVehicle(motorbike);
                    System.out.println(motorbike);
                } else {
                    System.out.println("Invalid option!! Re-enter.. ");
                    break;
                }
                break;
            }
        }else{
            System.out.println("Sorry, the parking is full!!");
        }
    }

    private void addVehicle(Vehicle vehicle) {
    }

    @Test
    public void deleteVehicle() {
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("|%1$-20s|%2$-20s|%3$-20s|\n","       TYPE","    PLATE
NUMBER","       MAKE ");
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("%1$-20s%2$-20s%3$-20s\n","","","");
        for(int i=0; i<listOfVehicles.size(); i++){
            System.out.format("|%1$-20s|%2$-20s|%3$-
20s|\n",listOfVehicles.get(i).getVehicleType(),"
"+listOfVehicles.get(i).getPlateNumber(),"  "+listOfVehicles.get(i).getMake());
        }
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
//          Scanner deleteVehicleInput = new Scanner(System.in);
        System.out.print("\n" +
                "Select vehicle type you want to delete" +
                "\n" +
                "1. Car \n" +
                "2. Motorbike \n" +
                "3. Main Menu\n" +
                "Choose: ");
        int deleteVehicleInput = 1;
        while (!(deleteVehicleInput==(int)deleteVehicleInput)){
            System.out.println("Invalid Data Type!!");
//              deleteVehicleInput.next();
```

```
            System.out.print("Select vehicle type you want to delete" +
                    "\n" +
                    "1. Car \n" +
                    "2. Motorbike \n"+
                    "3. Main Menu\n" +
                    "Choose: ");
        }
        int optionDeleteVehicle = 1;
        while (optionDeleteVehicle!=3) {
            if (optionDeleteVehicle == 1) {
                String vehicleType = "Car";
//                Scanner scannerDeleteVehicle = new Scanner(System.in);
                System.out.print("Enter the plate number of the vehicle that you
want to remove (WP ABC-1234): ");
                String deletePlateNumberInput = "WP CAA-2015"; //getting the plate
number to delete the vehicle.
                if (listOfVehicles.removeIf(removeVehicle ->
removeVehicle.getPlateNumber().equals(deletePlateNumberInput))) { //deleting the
vehicle.
                    System.out.println("Plate Number " + deletePlateNumberInput + ",
" + vehicleType + " has been removed.");
                    System.out.println(50-listOfVehicles.size() + " spaces are
available for parking");
                } else {
                    System.out.println("Plate number does not exists!!");//if the
plate number is not available, it will pop this message.
                }
            } else if (optionDeleteVehicle == 2) {
                String vehicleType = "Motorbike";
                Scanner scannerDeleteVehicle = new Scanner(System.in);
                System.out.print("Enter the plate number of th vehicle that you want
to remove (WP ABC-1234): ");
                String deletePlateNumberInput = "WP CCV*-2015"; //getting the plate
number to delete the vehicle.
                if (listOfVehicles.removeIf(removeVehicle ->
removeVehicle.getPlateNumber().equals(deletePlateNumberInput))) { //deleting the
vehicle.
                    System.out.println("Plate Number " + deletePlateNumberInput + ",
" + vehicleType + " has been removed.");
                    System.out.println(50-listOfVehicles.size() + " spaces are
available for parking");
                } else {
                    System.out.println("Plate number does not exists!!"); //if the
plate number is not available, it will pop this message.
                }
            } else {
                System.out.println("Invalid option!! Re-enter.. ");
            }
            break;
        }

    }

    @Test
    public void printVehicleList() {
        Collections.sort(listOfVehicles, new Comparator<Vehicle>() { //sorting the
vehicle list according to the vehicle make.
            @Override
            public int compare(Vehicle o1, Vehicle o2) {
                return String.valueOf(o1.getMake()).compareTo(o2.getMake());
            }
            @Override
            public boolean equals(Object obj) {
                return false;
            }
        });
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("|%1$-20s|%2$-20s|%3$-20s|\n","       TYPE","    PLATE 
NUMBER","        MAKE ");
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
        System.out.format("%1$-20s%2$-20s%3$-20s\n","","","");
        for(int i=0; i<listOfVehicles.size(); i++){
            System.out.format("|%1$-20s|%2$-20s|%3$-
20s|\n",listOfVehicles.get(i).getVehicleType()," 
"+listOfVehicles.get(i).getPlateNumber(),"  "+listOfVehicles.get(i).getMake());
        }
        System.out.format("_%1$-20s_%2$-20s_%3$-
20s\n","_____________________","_____________________","___________________");
    }

    @Test
    public void saveVehicleList() {
        File file = new File("VehicleList.txt");
        try{
            FileWriter fileWriter = new FileWriter(file);
            Writer output = new BufferedWriter(fileWriter);
            int size = listOfVehicles.size();

            for(int i=0; i<size; i++ ){
                output.write(listOfVehicles.get(i).toString()+ "\n");
            }output.close();

        }catch (Exception e){
            JOptionPane.showMessageDialog(null,"Cannot create the file.");
        }
        System.out.println("Successfully save to the file VehicleList.txt");
    }

    @Test
    public void saveInDatabase() {
    }
}
```

- Unit testing Snippets



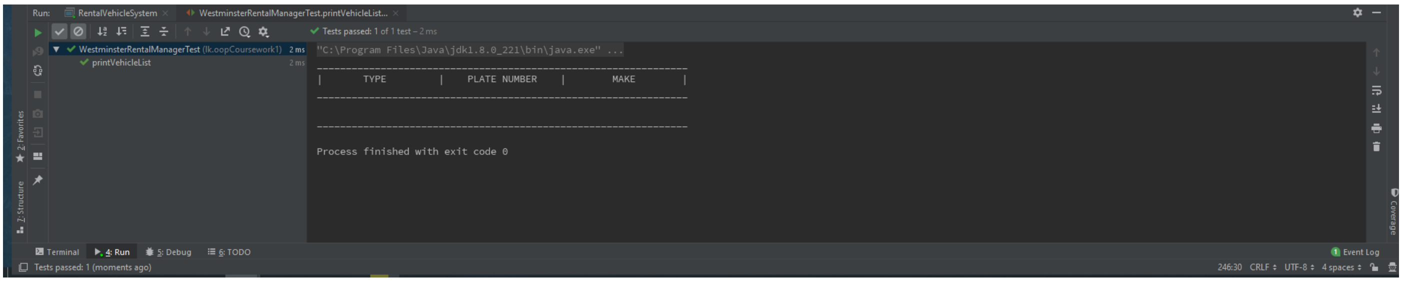
Run: RentalVehicleSystem | WestminsterRentalManagerTest.addVehicle1

✓ Tests passed: 1 of 1 test – 11 ms

✓ WestminsterRentalManagerTest (lk.oopCoursework1) 11 ms
✓ addVehicle1 11 ms

```
:: Motorbike{numberOfGears=5, wheelDiameter=16, numberOfHelmets=2} Vehicle{plateNumber='WP CCA-1414', make='Yamaha', numberOfPassengers=2, pricePerKm=30.0, vehicleType='Motorbike'}
```

Tests passed: 1 (moments ago)



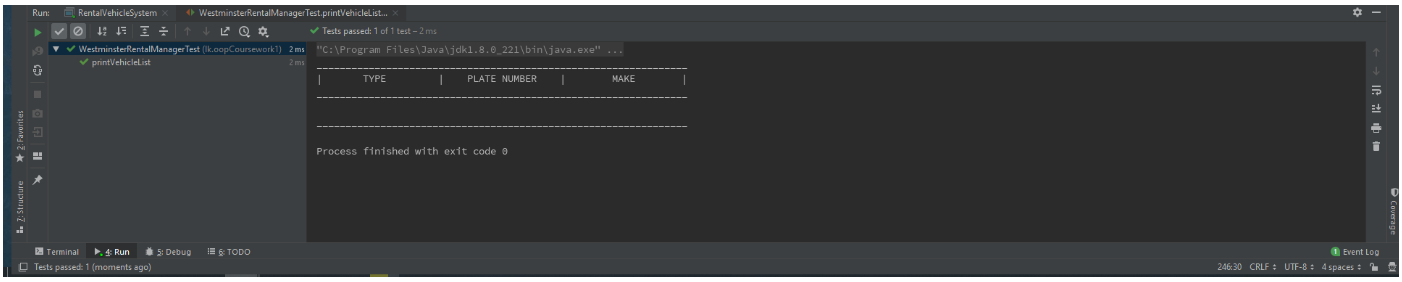
Run: RentalVehicleSystem | WestminsterRentalManagerTest.printVehicleList...

✓ Tests passed: 1 of 1 test – 2 ms

✓ WestminsterRentalManagerTest (lk.oopCoursework1) 2 ms
✓ printVehicleList 2 ms

```
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
----------------------------------------------------------------
|       TYPE         |    PLATE NUMBER    |        MAKE        |
----------------------------------------------------------------

----------------------------------------------------------------

Process finished with exit code 0
```

Tests passed: 1 (moments ago)



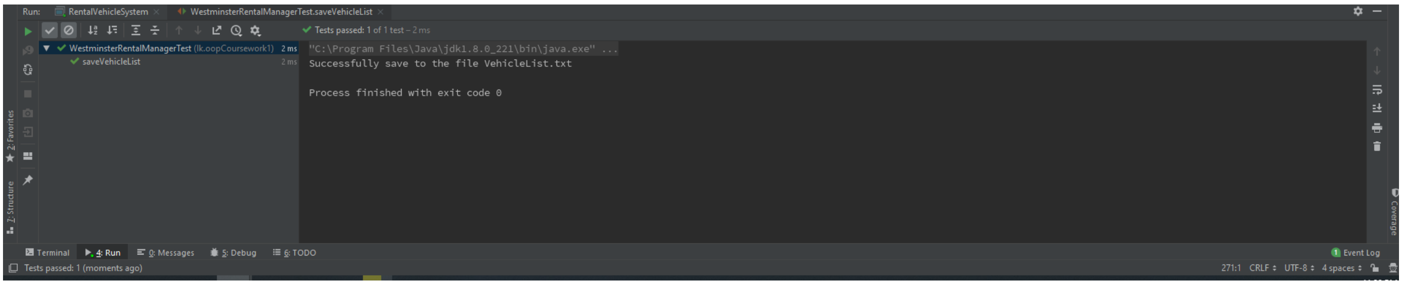
Run: RentalVehicleSystem | WestminsterRentalManagerTest.printVehicleList...

✓ Tests passed: 1 of 1 test – 2 ms

✓ WestminsterRentalManagerTest (lk.oopCoursework1) 2 ms
✓ printVehicleList 2 ms

```
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
----------------------------------------------------------------
|       TYPE         |    PLATE NUMBER    |        MAKE        |
----------------------------------------------------------------

----------------------------------------------------------------

Process finished with exit code 0
```

Tests passed: 1 (moments ago)



Run: RentalVehicleSystem | WestminsterRentalManagerTest.saveVehicleList

✓ Tests passed: 1 of 1 test – 2 ms

✓ WestminsterRentalManagerTest (lk.oopCoursework1) 2 ms
✓ saveVehicleList 2 ms

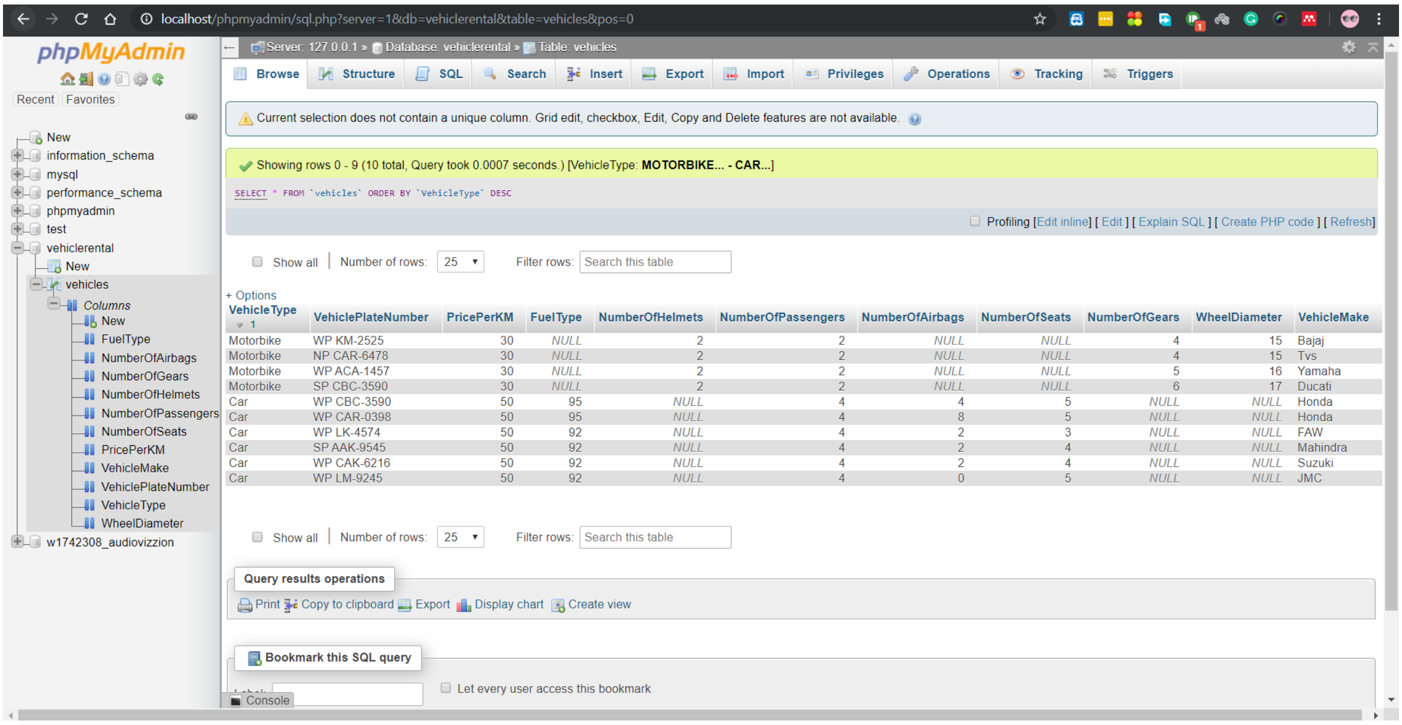
```
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" ...
Successfully save to the file VehicleList.txt

Process finished with exit code 0
```

Tests passed: 1 (moments ago)

❖ Other classes I have used

- Database connection class

```
package lk.oopCoursework1;

import java.sql.Connection;
import java.sql.DriverManager;

public class ConnectionClass {
    public Connection connection;

    public Connection getConnection(){
        String dbName = "vehiclerental";
        String userName = "root";
        String password = "";

        try{
            Class.forName("com.mysql.jdbc.Driver");
            connection =
DriverManager.getConnection("jdbc:mysql://localhost/"+dbName,userName,password);
        }catch (Exception exc){
            exc.printStackTrace();
        }
        return connection;
    }
}

//https://www.youtube.com/watch?v=NoPzqahrzp8&t=314s
```



Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 9 (10 total, Query took 0.0007 seconds.) [VehicleType: **MOTORBIKE... - CAR...**]

```
SELECT * FROM `vehicles` ORDER BY `VehicleType` DESC
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options

VehicleType ▾ 1	VehiclePlateNumber	PricePerKM	FuelType	NumberOfHelmets	NumberOfPassengers	NumberOfAirbags	NumberOfSeats	NumberOfGears	WheelDiameter	VehicleMake
Motorbike	WP KM-2525	30	NULL	2	2	NULL	NULL	4	15	Bajaj
Motorbike	NP CAR-6478	30	NULL	2	2	NULL	NULL	4	15	Tvs
Motorbike	WP ACA-1457	30	NULL	2	2	NULL	NULL	5	16	Yamaha
Motorbike	SP CBC-3590	30	NULL	2	2	NULL	NULL	6	17	Ducati
Car	WP CBC-3590	50	95	NULL	4	4	5	NULL	NULL	Honda
Car	WP CAR-0398	50	95	NULL	4	8	5	NULL	NULL	Honda
Car	WP LK-4574	50	92	NULL	4	2	3	NULL	NULL	FAW
Car	SP AAK-9545	50	92	NULL	4	2	4	NULL	NULL	Mahindra
Car	WP CAK-6216	50	92	NULL	4	2	4	NULL	NULL	Suzuki
Car	WP LM-9245	50	92	NULL	4	0	5	NULL	NULL	JMC

□ Show all | Number of rows: 25 ▾ Filter rows: Search this table

- RentalVehicleSystem (Main Method)

```
package lk.oopCoursework1;

import java.util.Scanner;

public class RentalVehicleSystem {

    public static void main(String[] args){
        WestminsterRentalManager rentsystem = new WestminsterRentalManager();
        Scanner mainMenuScanner = new Scanner(System.in);

        int menuOption=0;
        while (menuOption!=6){
            System.out.println(" \n" +
                    "      ______________________________________________ \n" +
                    "      Welcome to the Westminster Rental Vehicle Manager \n" +
                    "      ---------------------------------------------- \n" +
                    "\n" +
                    "1. Add vehicle \n" +
                    "2. Delete vehicle \n" +
                    "3. Print Vehicle list \n" +
                    "4. Save the vehicle list \n" +
                    "5. Open the gui \n"+
                    "6. Exit the programme \n" +
                    "\n");

            System.out.print("Choose an option: ");
            while (!mainMenuScanner.hasNextInt()){
                String wrongdatatype = mainMenuScanner.next();
                System.out.println("'"+wrongdatatype+"'"+" is an invalid data
type!!");
                System.out.print("Choose an option: ");
            }

            menuOption = mainMenuScanner.nextInt();
            switch (menuOption){
                case 1:
                    rentsystem.addVehicle();
                    break;
                case 2:
                    rentsystem.deleteVehicle();
                    break;
                case 3:
                    rentsystem.printVehicleList();
                    break;
                case 4:
                    rentsystem.saveVehicleList();
                    break;
                case 5:
                    rentsystem.saveInDatabase();
javafx.application.Application.launch(lk.oopCoursework1.Gui.class);
                    System.out.println("------->> Programme End <<--------");
                    System.exit(0);
                    break;
                case 6:
                    System.out.println("------->> Programme End <<--------");
                    rentsystem.saveInDatabase();
                    System.exit(0);
                default:
                    System.out.println("Please the choose the correct option!!");
            }
        }

    }
}
```

- Gui database connection

```
package lk.oopCoursework1;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class SqlConnection {


    public static Connection DbConnector(){
        String dbName = "vehiclerental";
        String userName = "root";
        String password = "";
        try{
            Connection conn = null;
            Class.forName("com.mysql.jdbc.Driver");
            conn =
DriverManager.getConnection("jdbc:mysql://localhost/"+dbName,userName,password);
            return conn;
        }catch (ClassNotFoundException | SQLException e){
            System.out.println(e);
        }
        return null;
    }
}

//https://www.youtube.com/watch?v=UD5Xf1z8Y-
4&list=PLeyMYhyx349ZZLdyNf1I7RODb83UwkJYo&index=14
```