

**Informatics Institute of Technology**

Department of Computing   
(B.Sc.) in Computer Science

**Module: 5COSC007C.1 Object Oriented Programming**

**Coursework 1**

**Phase 3**

Date : 18/11/2019

Student ID : 2018400

Student UoW ID : w1742308

Student First Name : Akila

Student Surname : Nanayakakra

Table of Contents

[List of vehicles in WestminsterRentalManager 3](#_Toc25011555)

[Screenshots 5](#_Toc25011556)

[Filter the vehicles 6](#_Toc25011557)

[Screenshots 7](#_Toc25011558)

[Full Code 8](#_Toc25011559)

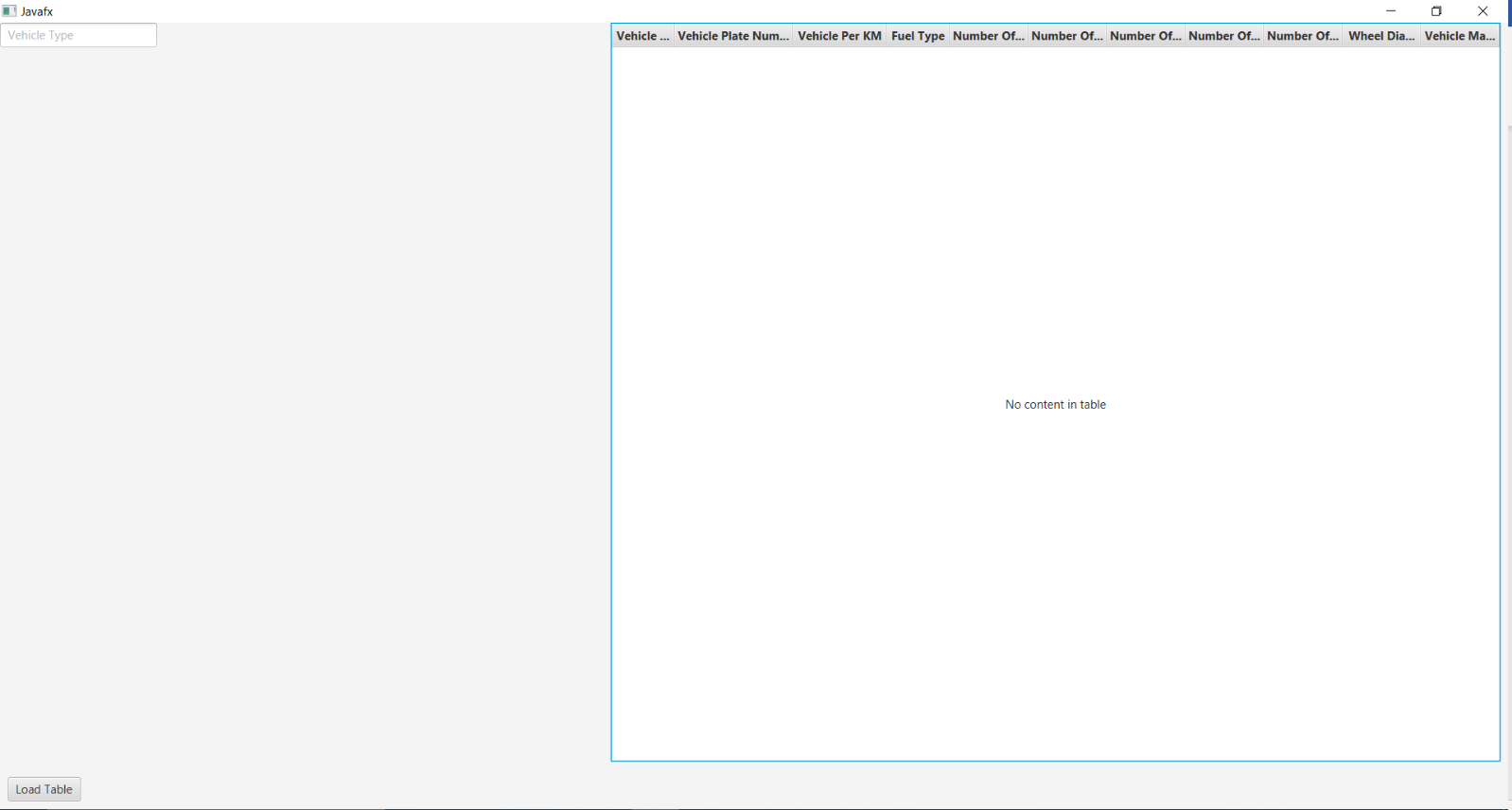
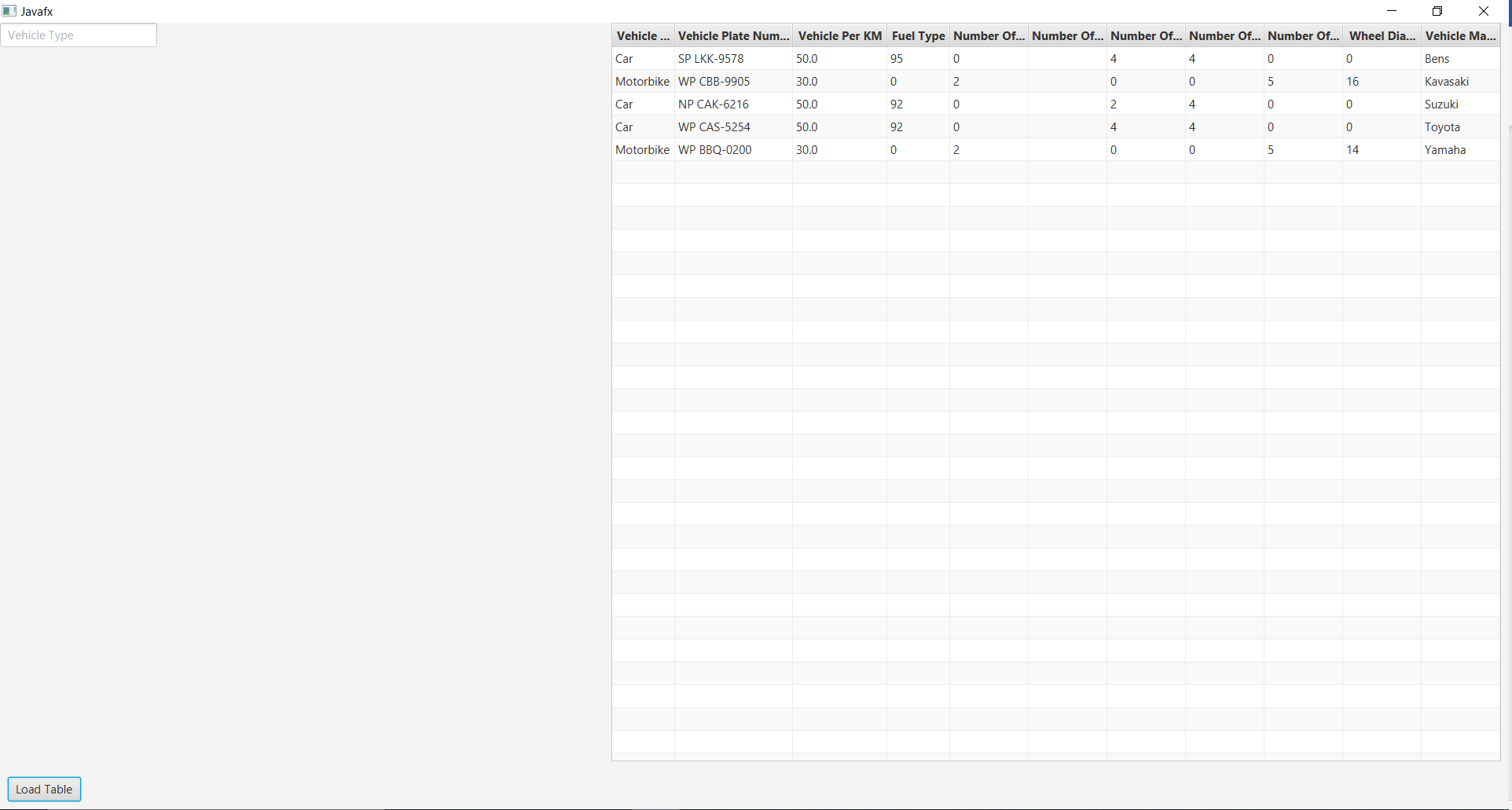
[User Class 11](#_Toc25011560)

[Connection 14](#_Toc25011561)

## List of vehicles in WestminsterRentalManager

@Override  
**public void** start(Stage primaryStage) **throws** Exception {  
 CheckConnection();  
  
 primaryStage.setTitle(**"Javafx"**);  
 BorderPane layout = **new** BorderPane();  
 Scene newScene = **new** Scene(layout,1000,600,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,column8,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));  
  
   
 primaryStage.setScene(newScene);  
 primaryStage.show();  
  
}

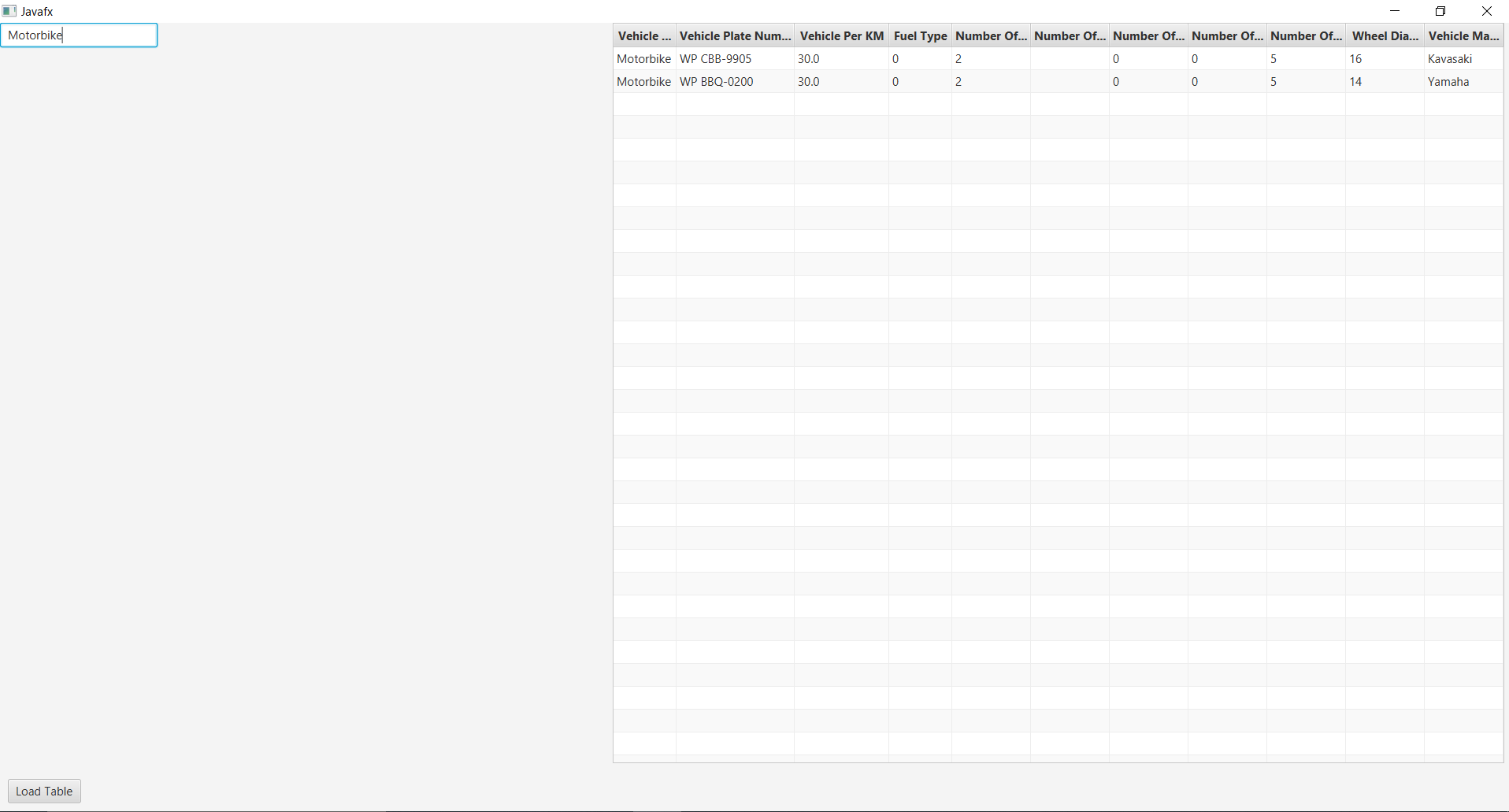
### Screenshots

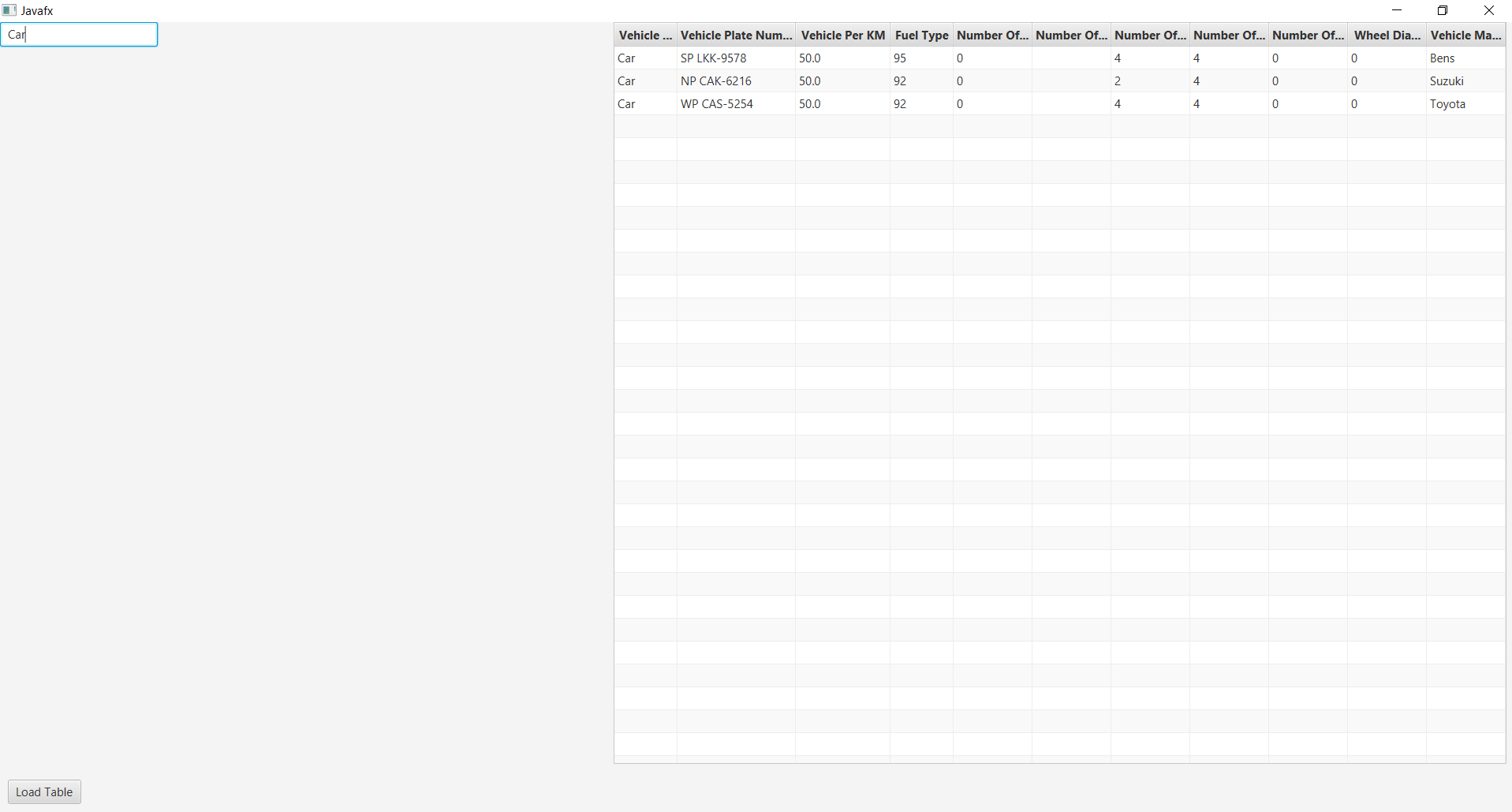
* This is the first page. We should load it to get the data from the data base. When we click on load table, it shows the data.
* Above table shows the list of vehicles in the Westminster rental manager. It has taken from the data base.

## Filter the vehicles

@Override  
**public void** start(Stage primaryStage) **throws** Exception {  
 CheckConnection();  
  
 primaryStage.setTitle(**"Javafx"**);  
 BorderPane layout = **new** BorderPane();  
 Scene newScene = **new** Scene(layout,1000,600,Color.*rgb*(0,0,0,0));  
  
 TableView<User> table = **new** TableView<>();  
 **final** ObservableList<User> data = FXCollections.*observableArrayList*();  
  
   
 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();  
  
}

### Screenshots

* I filter the vehicles according to the type of the vehicle, car or motorbike.



## Full Code

**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,1000,600,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,column8,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();  
  
 }  
}

## User Class

**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** getNumberPfPassengers() {  
 **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** setNumberPfPassengers(**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);  
 }  
}

## 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**;  
 }  
}