EAD Assignment

EAD

Year 4, Semester 2, 2022

Group ID: 57

Topic - Mobile Application to Manage Fuel Queues in Fuel Stations Using Client-Server Architecture

GitHub link:

https://github.com/lakindulk/Fuel_app

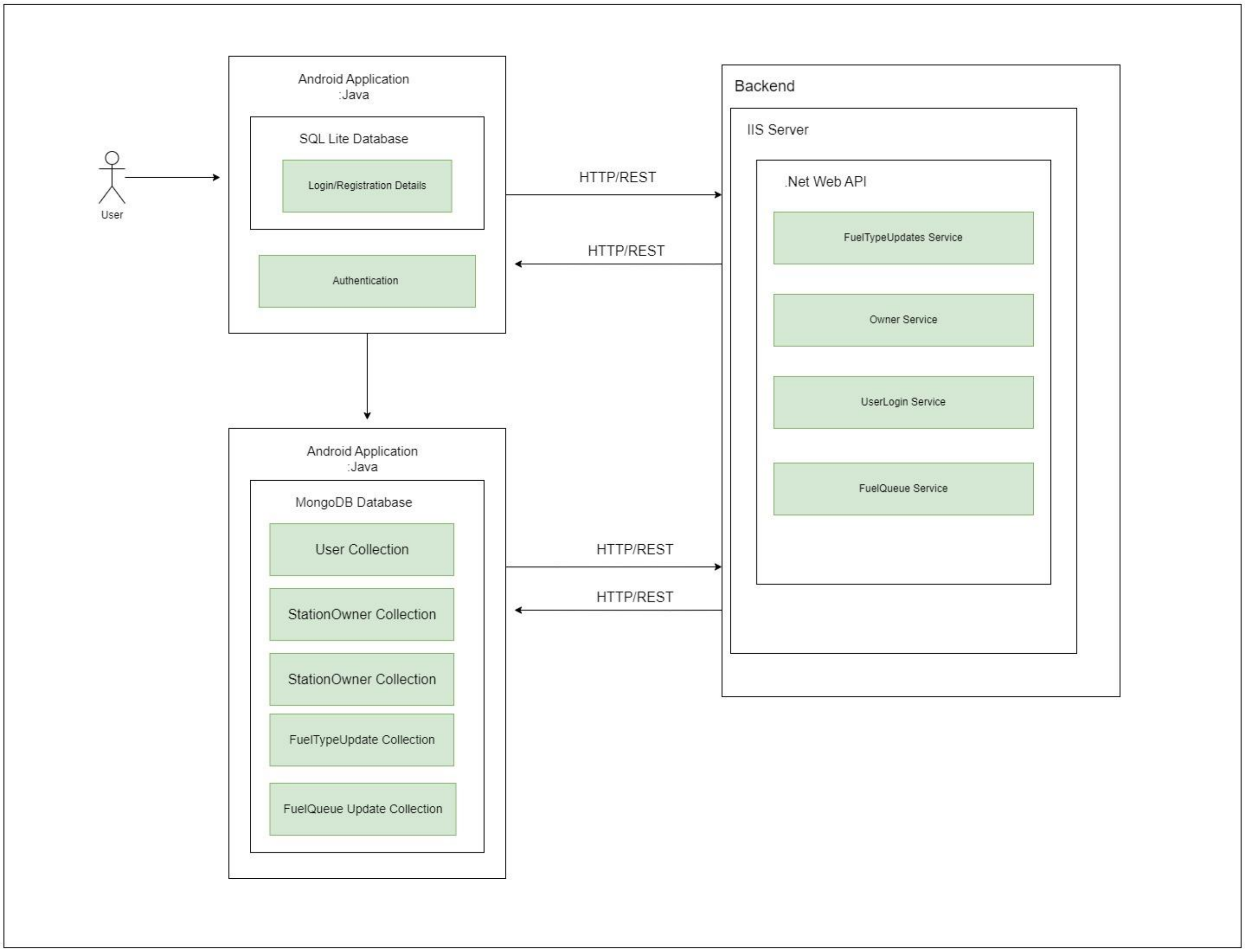
https://github.com/lakindulk/Backend_Fuel_app

Name	Registration Number
Wishwajith W. L	IT19140230
Kavishka K.A. L	IT19122656
Jayasekara D.D.K. G	IT19220048

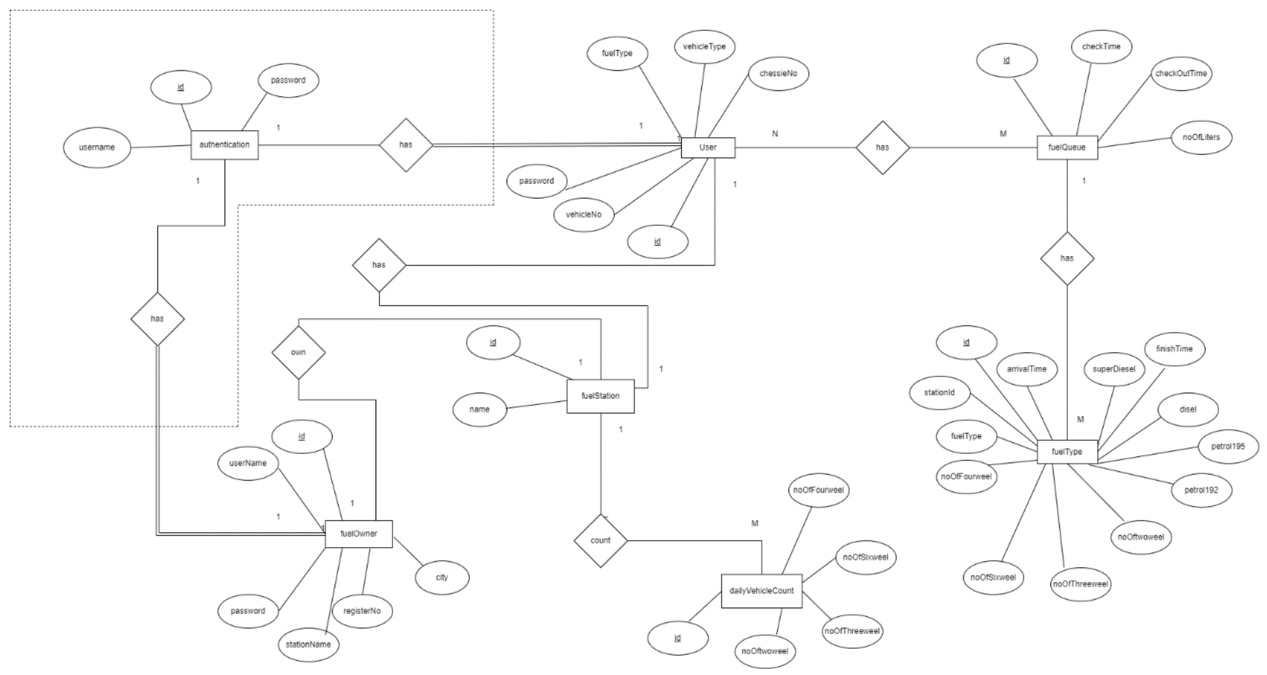
NOTE: We have copied our code below. It’s not screenshots. Because of the dark theme, it looks like a screenshot.

Table of Contents

1.Application High Level Diagram

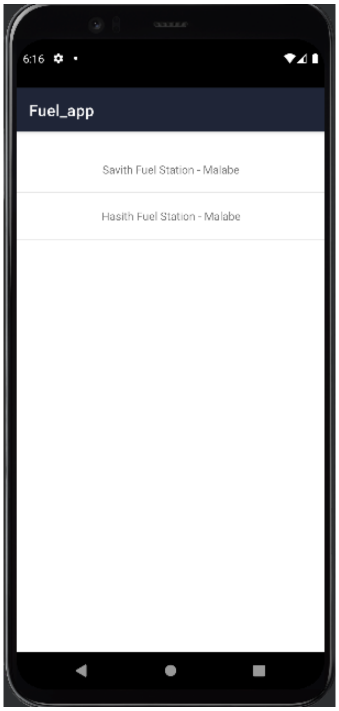


2.Database Design



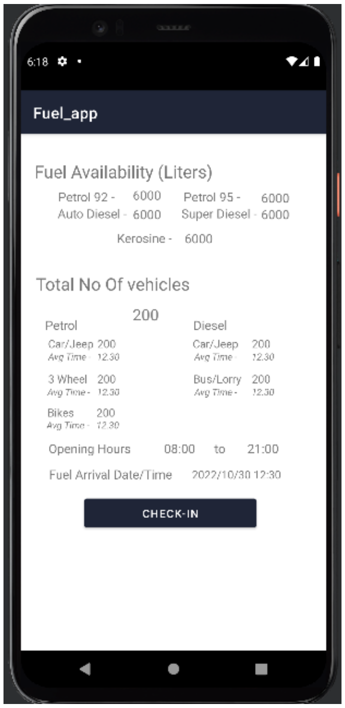
3.ScreenShots of UI’s

Wishwajith W.L (IT19140230)



Savith Fuel Station - Malabe

Hasith Fuel Station - Malabe



Fuel Availability (Liters)

Petrol 92 - 6000 Petrol 95 - 6000
Auto Diesel - 6000 Super Diesel - 6000
Kerosine - 6000

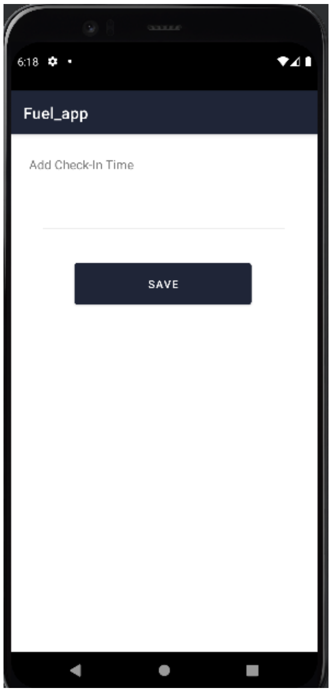
Total No Of vehicles

200

Petrol		Diesel	
Car/Jeep	200	Car/Jeep	200
Avg Time -	*12.30*	*Avg Time -*	*12.30*
3 Wheel	200	Bus/Lorry	200
Avg Time -	*12.30*	*Avg Time -*	*12.30*
Bikes	200		
Avg Time -	*12.30*		

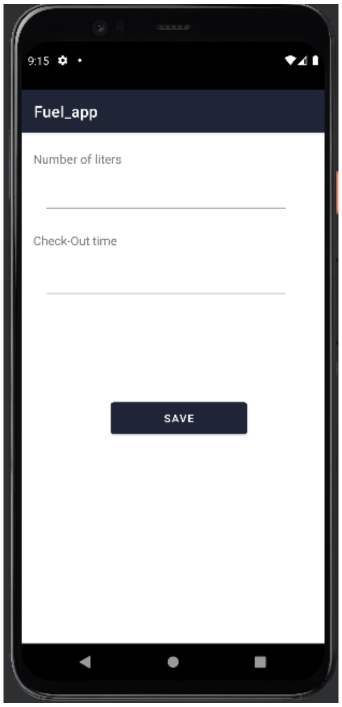
Opening Hours 08:00 to 21:00

Fuel Arrival Date/Time 2022/10/30 12:30



Add Check-In Time

Add Check out and

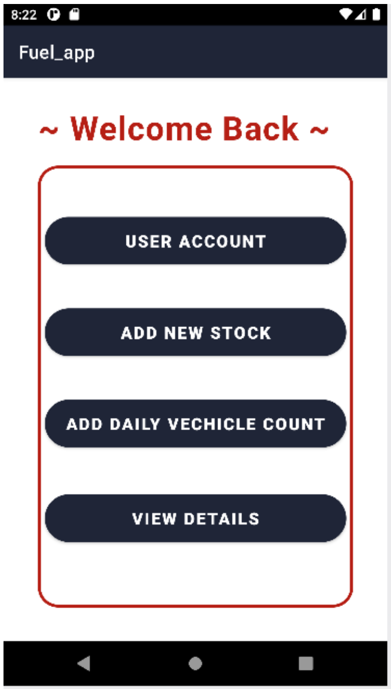


Number of liters

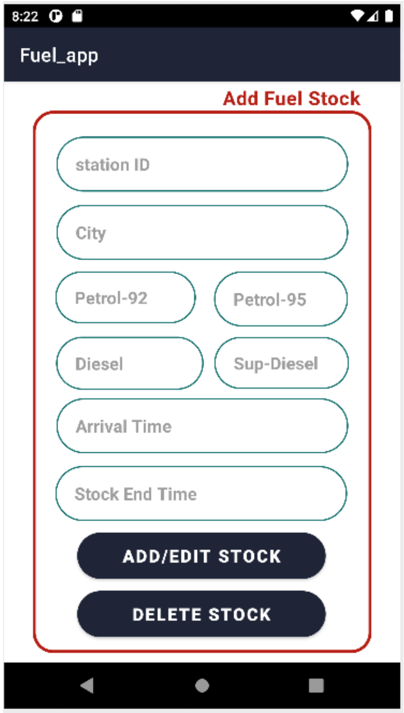
Check-Out time

Kavishka K.A.L (IT19122656)

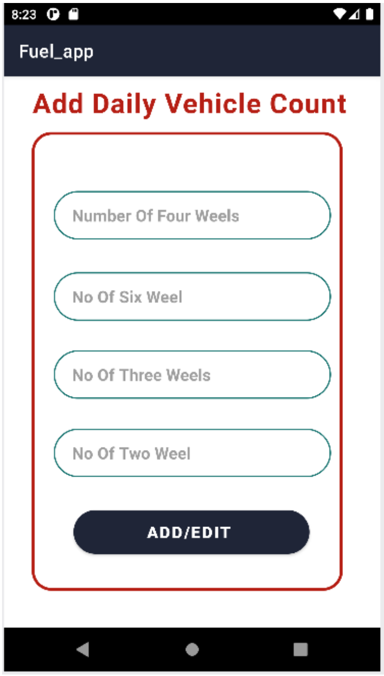
Station Owner Landing Page



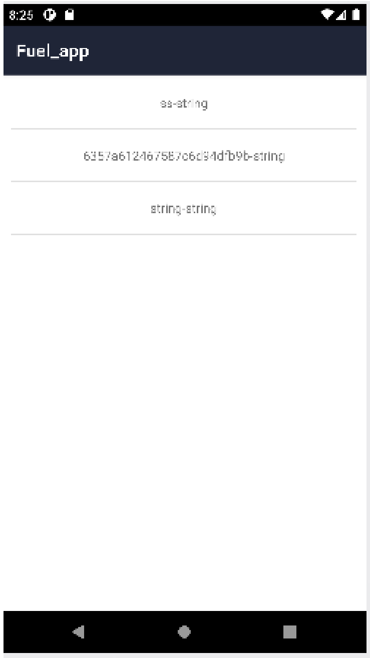
Add New Stock UI



Add Daily Vehicle Count



View Details

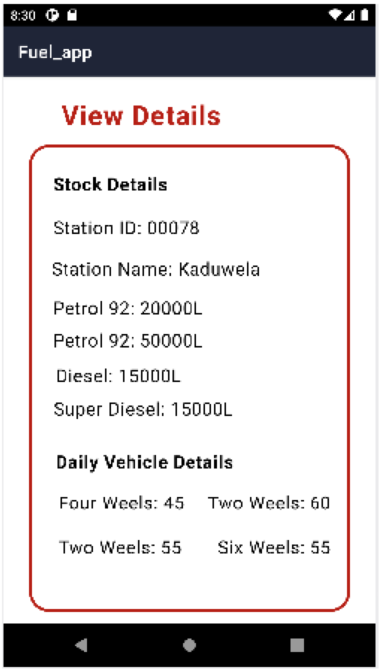


ss-string

6357a612467587c6d94dfb9b-string

string-string

View Single Item Details



Stock Details

Station ID: 00078

Station Name: Kaduwela

Petrol 92: 20000L

Petrol 92: 50000L

Diesel: 15000L

Super Diesel: 15000L

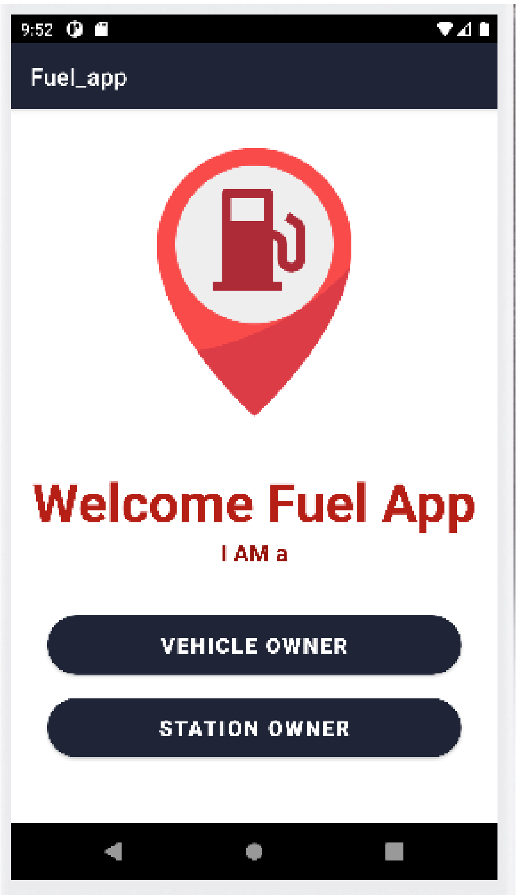
Daily Vehicle Details

Four Weels: 45 Two Weels: 60

Two Weels: 55 Six Weels: 55

Jayasekara D.D.K.G (IT19220048)

Splash Screen

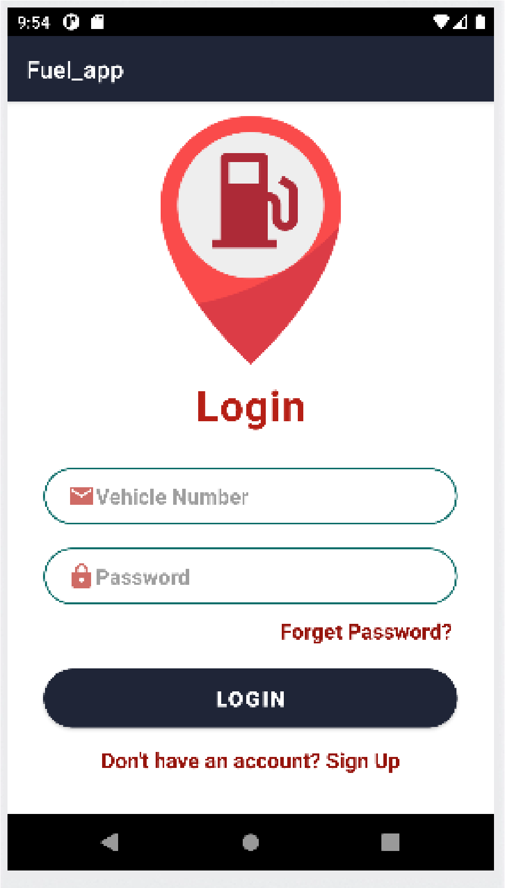


Fuel_app

Welcome Fuel App

I AM a

Vehicle Owner Login Screen



Fuel_app

Login

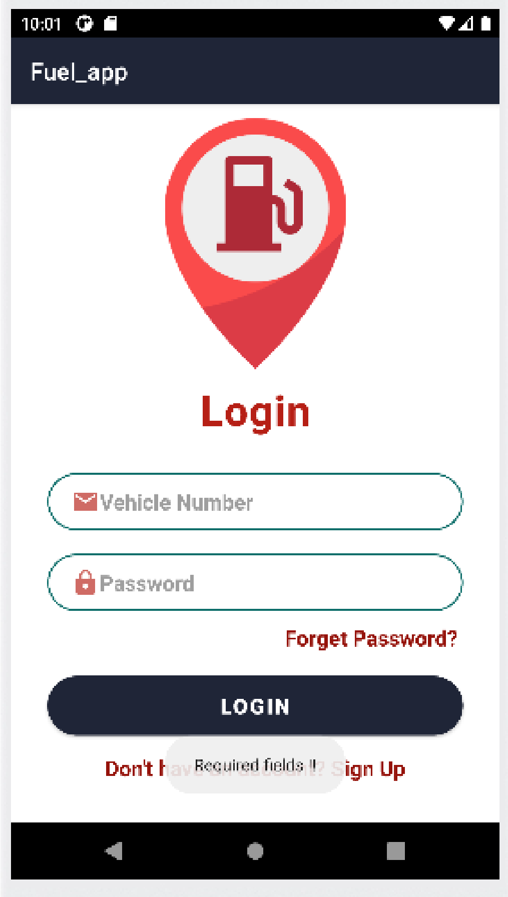
After clicks vehicle owner button vehicle owner login screen will open.

After clicks Station owner button station owner login screen will open.

A vehicle owner should add valid login credentials for login to the app. If not displays error message.

If user is not registered to the app, can register by clicking sign up.

Error message

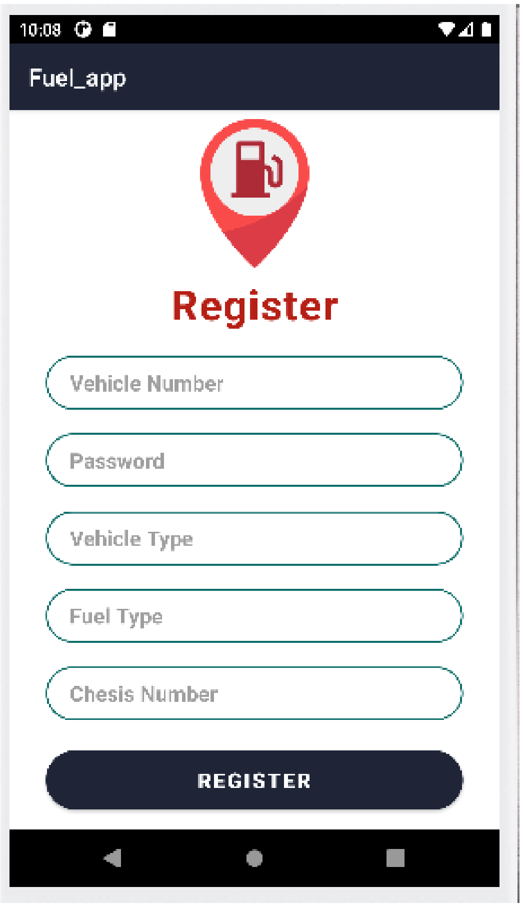


Required fields !!

If user tries to go next page without entering details he/she can see error message.

After enter valid login credentials user can login to the system. Successful Toast message can see user **(Feature)**

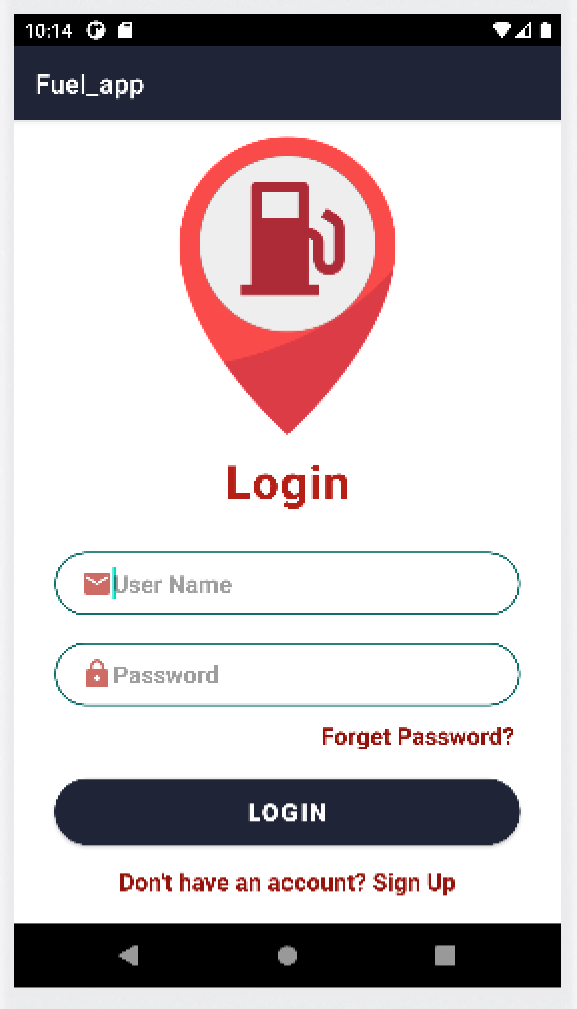
Vehicle Owner Registration Screen

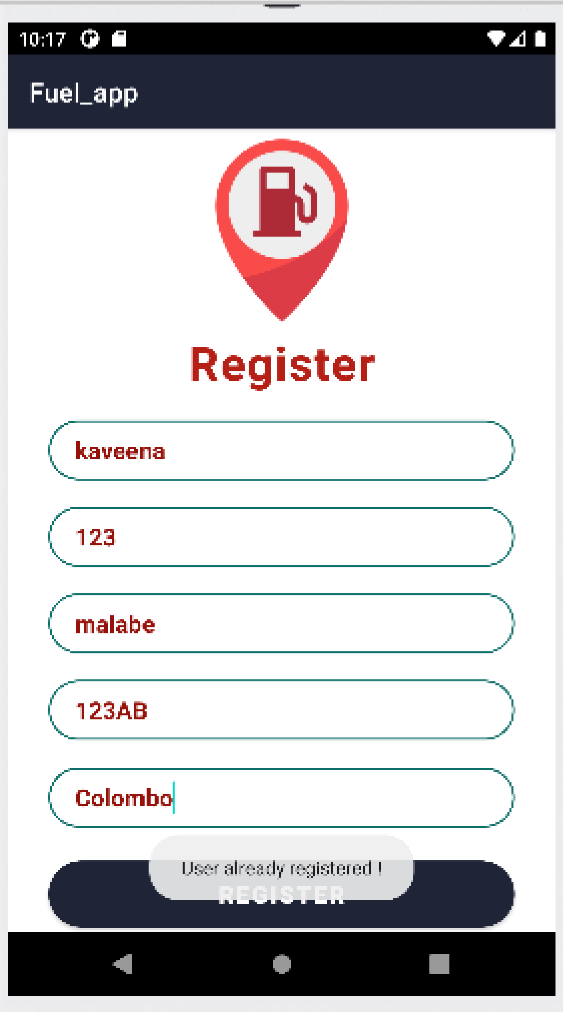


If user tries to go next page without entering one of details he/she can see error messages and can not register to the app.

Station Owner Login Screen

Station Owner Registration screen





kaveena

123

malabe

123AB

Colombo

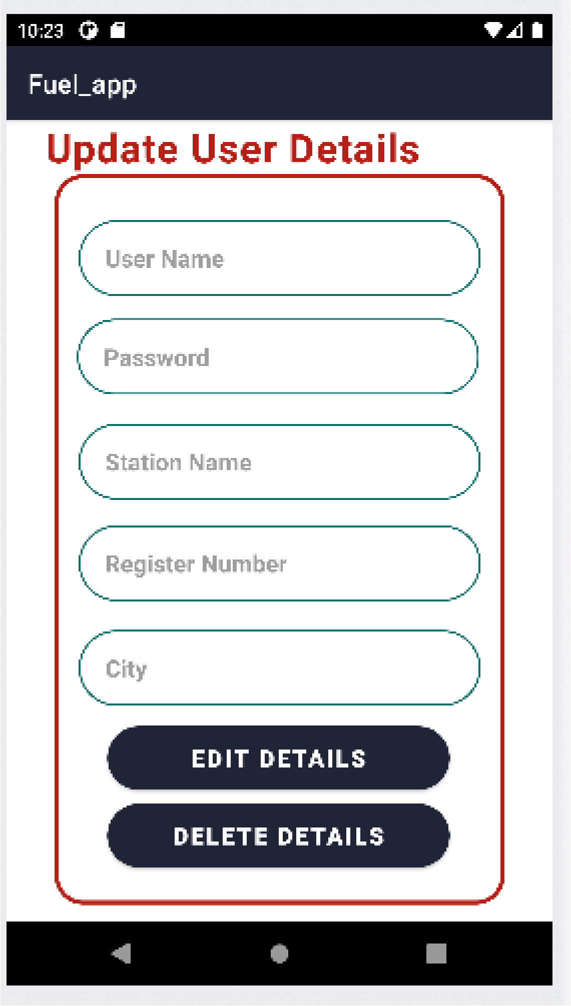
User already registered !

A station owner should add valid login credentials for login to the app. If not displays error message.

If owner is not registered to the app, can register by clicking sign up.

Also if user already register to the system, he/she can't register again. System will display above toast message and can't continue.

User Management UI



Also if user can edit their existing details in this UI and if he/she want to delete details can delete details by clicking delete details button

4.Code

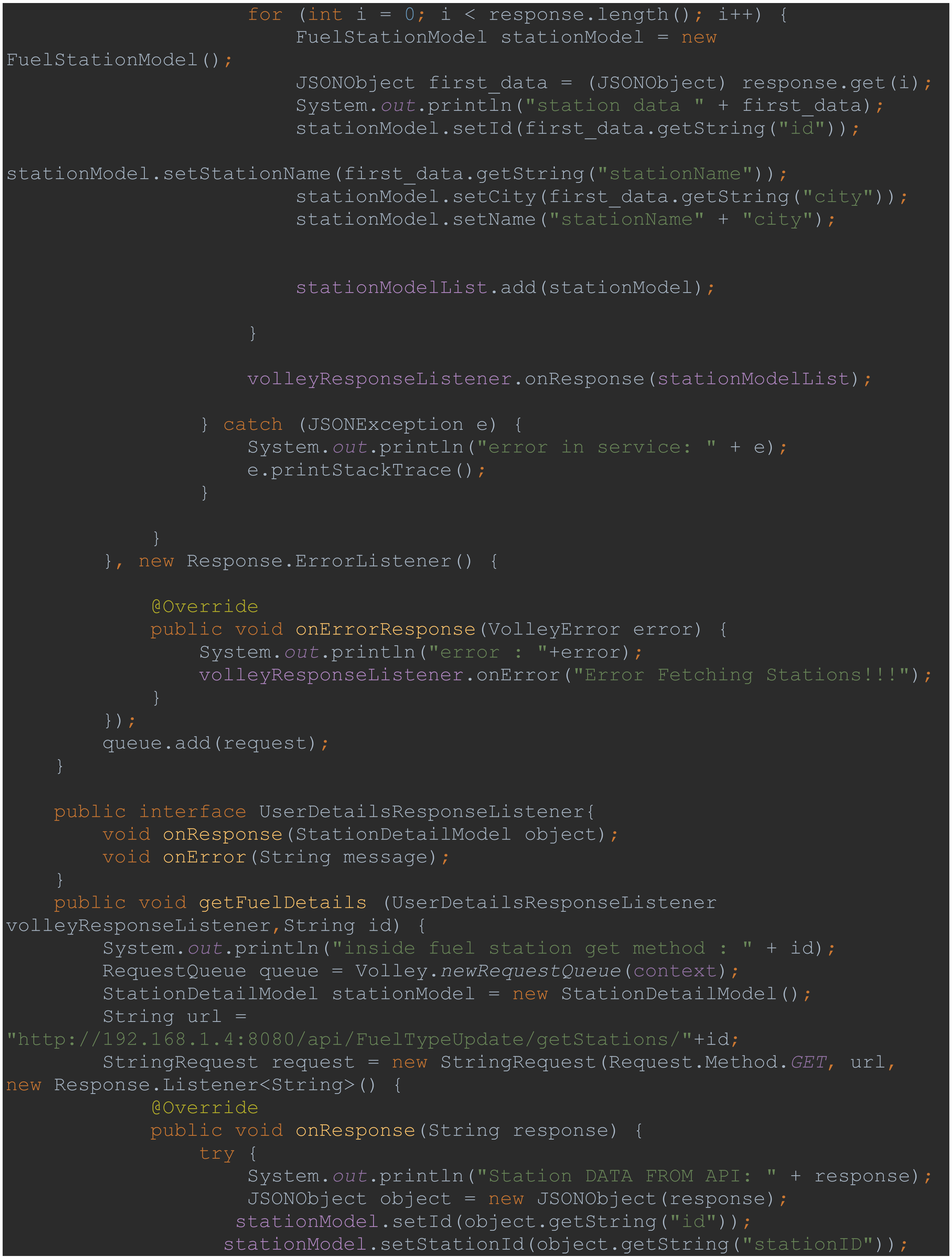
4.1 Frontend –

Wishwajith W.L – IT19140230

JAVA classes

StationDetailsService.java

```
package com.example.fuel_app;

import android.content.Context;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class StationDetailsService {
    Context context;

    public StationDetailsService(Context context) {
        this.context = context;
    }

    public interface VolleyResponseListener{
        void onResponse(ArrayList<FuelStationModel> stationModel);
        void onError(String message);
    }



    public void getAllStations(VolleyResponseListener volleyResponseListener)
{
        System.out.println("inside get all");
        ArrayList<FuelStationModel> stationModelList = new ArrayList<>();
        RequestQueue queue = Volley.newRequestQueue(context);
        String url = "http://192.168.1.4:8080/api/Owner";
        JsonArrayRequest request = new JsonArrayRequest(Request.Method.GET,
url, null, new Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {
                try {
                    System.out.println("Hi: " + response);
```



```
                    for (int i = 0; i < response.length(); i++) {
                        FuelStationModel stationModel = new
FuelStationModel();
                        JSONObject first_data = (JSONObject) response.get(i);
                        System.out.println("station data " + first_data);
                        stationModel.setId(first_data.getString("id"));

stationModel.setStationName(first_data.getString("stationName"));
                        stationModel.setCity(first_data.getString("city"));
                        stationModel.setName("stationName" + "city");


                        stationModelList.add(stationModel);

                    }

                    volleyResponseListener.onResponse(stationModelList);

                } catch (JSONException e) {
                    System.out.println("error in service: " + e);
                    e.printStackTrace();
                }

            }
        }, new Response.ErrorListener() {

            @Override
            public void onErrorResponse(VolleyError error) {
                System.out.println("error : "+error);
                volleyResponseListener.onError("Error Fetching Stations!!!");
            }
        });
        queue.add(request);
    }

    public interface UserDetailsResponseListener{
        void onResponse(StationDetailModel object);
        void onError(String message);
    }
    public void getFuelDetails (UserDetailsResponseListener
volleyResponseListener,String id) {
        System.out.println("inside fuel station get method : " + id);
        RequestQueue queue = Volley.newRequestQueue(context);
        StationDetailModel stationModel = new StationDetailModel();
        String url =
"http://192.168.1.4:8080/api/FuelTypeUpdate/getStations/"+id;
        StringRequest request = new StringRequest(Request.Method.GET, url,
new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    System.out.println("Station DATA FROM API: " + response);
                    JSONObject object = new JSONObject(response);
                   stationModel.setId(object.getString("id"));
                  stationModel.setStationId(object.getString("stationID"));
```

```
                    stationModel.setDiesel(object.getString("diesel"));
                   stationModel.setPetrol92(object.getString("petrol92"));
                stationModel.setPetrol95(object.getString("petrol95"));

stationModel.setSuperDiesel(object.getString("superDiesel"));

stationModel.setArrivalTime(object.getString("arrivalTime"));
                } catch (JSONException e) {
                    e.printStackTrace();
                }

            }

        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                volleyResponseListener.onError("Error Fetching
User!!!"+error);
            }
        });
        queue.add(request);


    }




}
```

CustomBaseAdapter.js

```
package com.example.fuel_app;

import android.annotation.SuppressLint;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.BaseAdapter;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

import java.util.ArrayList;

public class CustomBaseAdapter extends ArrayAdapter<FuelStationModel> {

    //Reference:
https://www.youtube.com/watch?v=RHqGiWluAzU&ab_channel=Foxandroid
    public CustomBaseAdapter(@NonNull Context context,
ArrayList<FuelStationModel> stationArrayList) {
        super(context, R.layout.activity_custom_list_view,stationArrayList);


    }

    @SuppressLint("ResourceAsColor")
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent) {
        System.out.println("inside get view");
        FuelStationModel model = getItem(position);

        if(convertView == null){
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.activity_custom_list_view,
parent,false);
        }

        System.out.println("station Name : " + model.getName());
        TextView stationName = convertView.findViewById(R.id.textView);

        stationName.setText(model.getStationName() + " - " +model.getCity());

        return convertView;
      //return super.getView(position, convertView, parent);
    }
}
```

ViewFuelStation.java

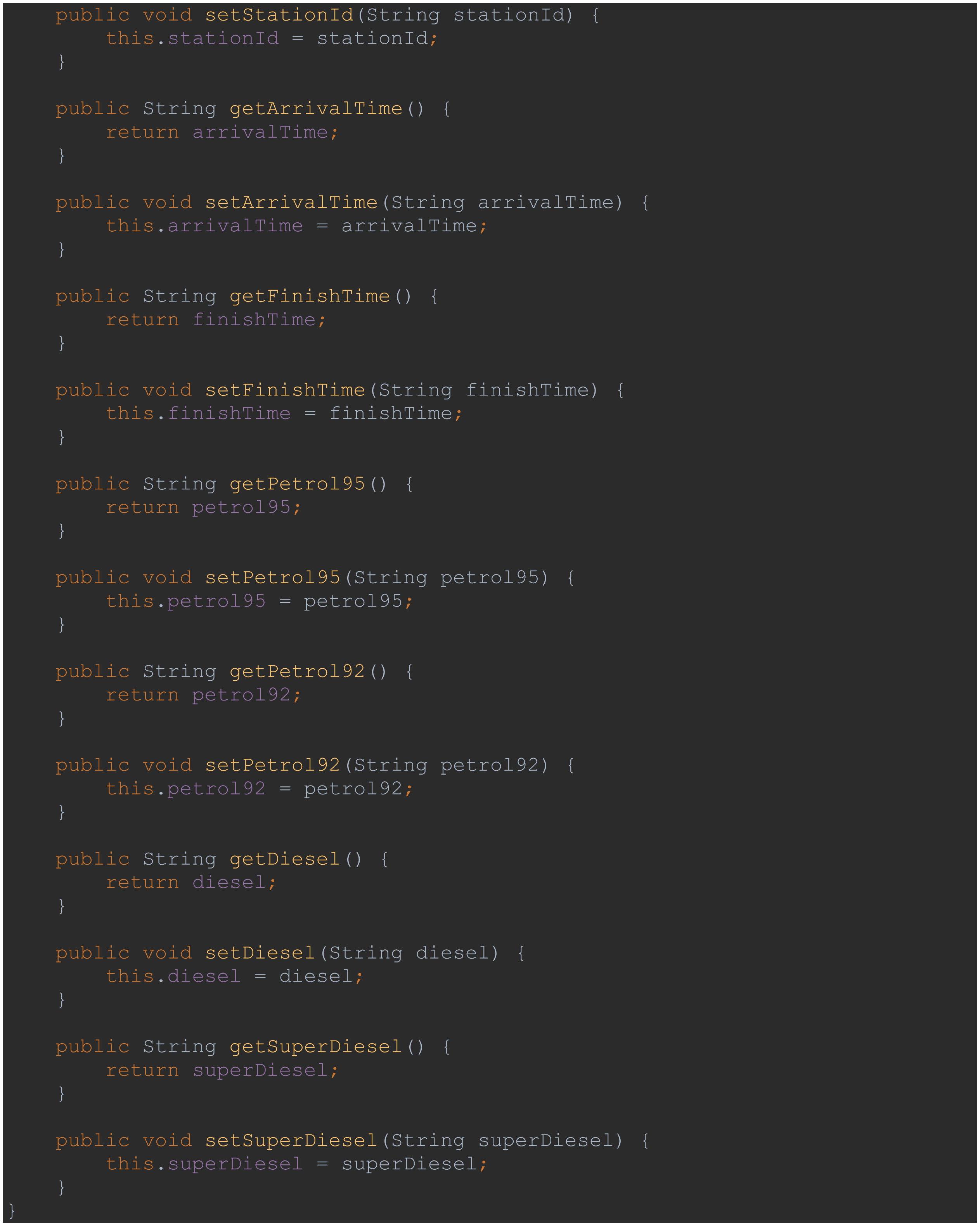
```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;

public class ViewFuelStation extends AppCompatActivity {

    ListView stationList;
    CustomBaseAdapter listAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_fuel_station);
        stationList = (ListView) findViewById(R.id.customListView);

        StationDetailsService stationDataService = new
StationDetailsService(ViewFuelStation.this);
        stationDataService.getAllStations(new
StationDetailsService.VolleyResponseListener() {
            @Override
            public void onResponse(ArrayList<FuelStationModel> stationModel)
{
                System.out.println("station data : " + stationModel);
//                ArrayAdapter arr = new ArrayAdapter(ViewFuelStation.this,
android.R.layout.simple_list_item_1,stationModel.get());

                listAdapter = new
CustomBaseAdapter(getApplicationContext(),stationModel);
                stationList.setAdapter(listAdapter);
                stationList.setClickable(true);
                stationList.setOnItemClickListener(new
AdapterView.OnItemClickListener(){

                    @Override
                    public void onItemClick(AdapterView<?> adapterView, View
view, int i, long l) {
                        System.out.println("Clicked One:" + i);
                        Intent intent = new
Intent(getApplicationContext(),Queue_home.class);
                        intent.putExtra("ID",stationModel.get(i).getId());
                        startActivity(intent);
                    }
                });
            }

            @Override
            public void onError(String message) {
                Toast.makeText(ViewFuelStation.this,message ,
Toast.LENGTH_LONG).show();
            }
        });

    }
}
```

Queue_home.java

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;

public class Queue_home extends AppCompatActivity {
    Button checkIn;
    String id;
    TextView pet92, pet95, die, dieS,kero,arrTime;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_queue_home);

        checkIn = findViewById(R.id.checkInBtn);
        pet92 = findViewById(R.id.petrol92Val);
        pet95 = findViewById(R.id.petrol95Val);
        die = findViewById(R.id.autodTxt);
        dieS = findViewById(R.id.superDVal);
        kero = findViewById(R.id.keroVal);
        arrTime = findViewById(R.id.arivalTime);

        Intent intent = this.getIntent();
        if(intent != null){
            id = intent.getStringExtra("ID");
        }
        System.out.println("id : " + id);

        StationDetailsService stationDataService = new
StationDetailsService(Queue_home.this);
        stationDataService.getFuelDetails(new
StationDetailsService.UserDetailsResponseListener() {
            @Override
            public void onResponse(StationDetailModel stationModel) {
                System.out.println("station data : " + stationModel);
//                  ArrayAdapter arr = new ArrayAdapter(ViewFuelStation.this,
android.R.layout.simple_list_item_1,stationModel.get());
                    pet92.setText(stationModel.getPetrol92());
                    pet95.setText(stationModel.getPetrol95());
                    die.setText(stationModel.getDiesel());
                    dieS.setText(stationModel.getSuperDiesel());
                    arrTime.setText(stationModel.getArrivalTime());
            }

            @Override
            public void onError(String message) {
                Toast.makeText(Queue_home.this,message ,
Toast.LENGTH_LONG).show();
            }
        },id);
        checkIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new
Intent(Queue_home.this,Add_checkin_time.class);
                intent.putExtra("ID",id);
                startActivity(intent);
            }
        });
    }
}
```

StationModelService.java

```
package com.example.fuel_app;

public class StationDetailModel {
    private String id;
    private String stationId;
    private String arrivalTime;
    private String finishTime;
    private String petrol95;
    private String petrol92;
    private String diesel;
    private String superDiesel;

    public StationDetailModel() {
    }

    public StationDetailModel(String id, String stationId, String
arrivalTime, String finishTime, String petrol95, String petrol92, String
diesel, String superDiesel) {
        this.id = id;
        this.stationId = stationId;
        this.arrivalTime = arrivalTime;
        this.finishTime = finishTime;
        this.petrol95 = petrol95;
        this.petrol92 = petrol92;
        this.diesel = diesel;
        this.superDiesel = superDiesel;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getStationId() {
        return stationId;
    }
```



```java
    public void setStationId(String stationId) {
        this.stationId = stationId;
    }

    public String getArrivalTime() {
        return arrivalTime;
    }

    public void setArrivalTime(String arrivalTime) {
        this.arrivalTime = arrivalTime;
    }

    public String getFinishTime() {
        return finishTime;
    }

    public void setFinishTime(String finishTime) {
        this.finishTime = finishTime;
    }

    public String getPetrol95() {
        return petrol95;
    }

    public void setPetrol95(String petrol95) {
        this.petrol95 = petrol95;
    }

    public String getPetrol92() {
        return petrol92;
    }

    public void setPetrol92(String petrol92) {
        this.petrol92 = petrol92;
    }

    public String getDiesel() {
        return diesel;
    }

    public void setDiesel(String diesel) {
        this.diesel = diesel;
    }

    public String getSuperDiesel() {
        return superDiesel;
    }

    public void setSuperDiesel(String superDiesel) {
        this.superDiesel = superDiesel;
    }
}
```

Add_checkin_time.java

```java
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.text.Editable;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;

public class Add_checkin_time extends AppCompatActivity {

    Editable checkInTime;
    Button checkInBtn;
    EditText cheinInTxt;
    private RequestQueue requestQueue;
    String id;

    @SuppressLint({"WrongViewCast", "MissingInflatedId"})
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_checkin_time);

        cheinInTxt = findViewById(R.id.addCheckInTime);
        checkInBtn = findViewById(R.id.addTime);

        Intent intent = this.getIntent();
        if(intent != null){
            id = intent.getStringExtra("ID");
        }
        System.out.println("id : " + id);


        checkInBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                checkInTime = cheinInTxt.getText();

                if(checkInTime.equals("") ){
                    Toast.makeText(Add_checkin_time.this, "Enter Checkin 
time", Toast.LENGTH_SHORT).show();

                } else {
                    try {
                        // url to post the user data
                        String url = 
"http://192.168.1.4:8080/api/FuelQueueUpdate";

                        HashMap<String, String> params = new HashMap<String, 
String>();

                        params.put("StationId", id);
                        params.put("VehicleType", "Car");
                        params.put("NoOfLiters", "");
                        params.put("CheckInTime", checkInTime.toString());
                        params.put("CheckOutTime", "");
                        params.put("FuelType", "Petrol");

                        JsonObjectRequest req = new JsonObjectRequest(url, 
new JSONObject(params),
                                new Response.Listener<JSONObject>() {
                                    @Override
                                    public void onResponse(JSONObject 
response) {
                                        try {
                                            VolleyLog.v("Response:%n %s", 
response.toString(4));
                                        } catch (JSONException e) {
                                            e.printStackTrace();
                                        }
                                    }
                                }, new Response.ErrorListener() {
                            @Override
                            public void onErrorResponse(VolleyError error) {
                                VolleyLog.e("Error: ", error.getMessage());
                            }
                        });

                        requestQueue = 
Volley.newRequestQueue(Add_checkin_time.this);
                        requestQueue.add(req);

                        Intent intent = new 
Intent(Add_checkin_time.this,Add_checkout_time.class);
                        startActivity(intent);

                    }catch (Exception e){

                    }
                }



            }
        });
```

```
    }
}
```

FuelStationModel.java

```
package com.example.fuel_app;

public class FuelStationModel {
    private String id;
    private String stationName;
    private String city;
    private String name;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public FuelStationModel(String id, String stationName, String city) {
        this.id = id;
        this.stationName = stationName;
        this.city = city;

    }

    public FuelStationModel() {
    }

    @Override
    public String toString() {
        return "FuelStationModel{" +
                "id='" + id + '\'' +
                ", stationName='" + stationName + '\'' +
                ", city='" + city + '\'' +

                '}';
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getStationName() {
        return stationName;
    }
    public void setStationName(String stationName) {
        this.stationName = stationName;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }


}
```

Kavishka K.A.L (IT19122656)

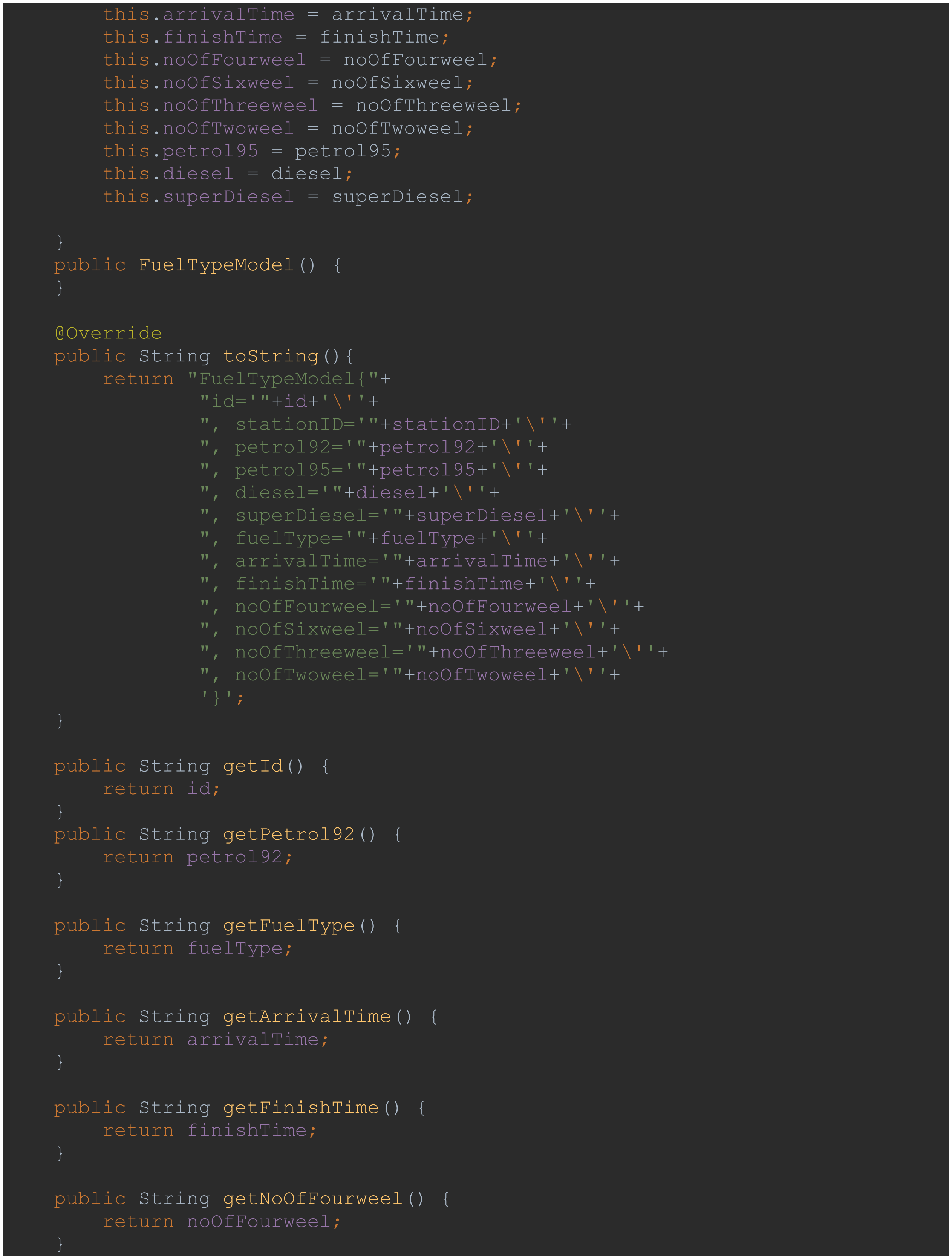
JAVA Classes

FueltypeMainActivity.class

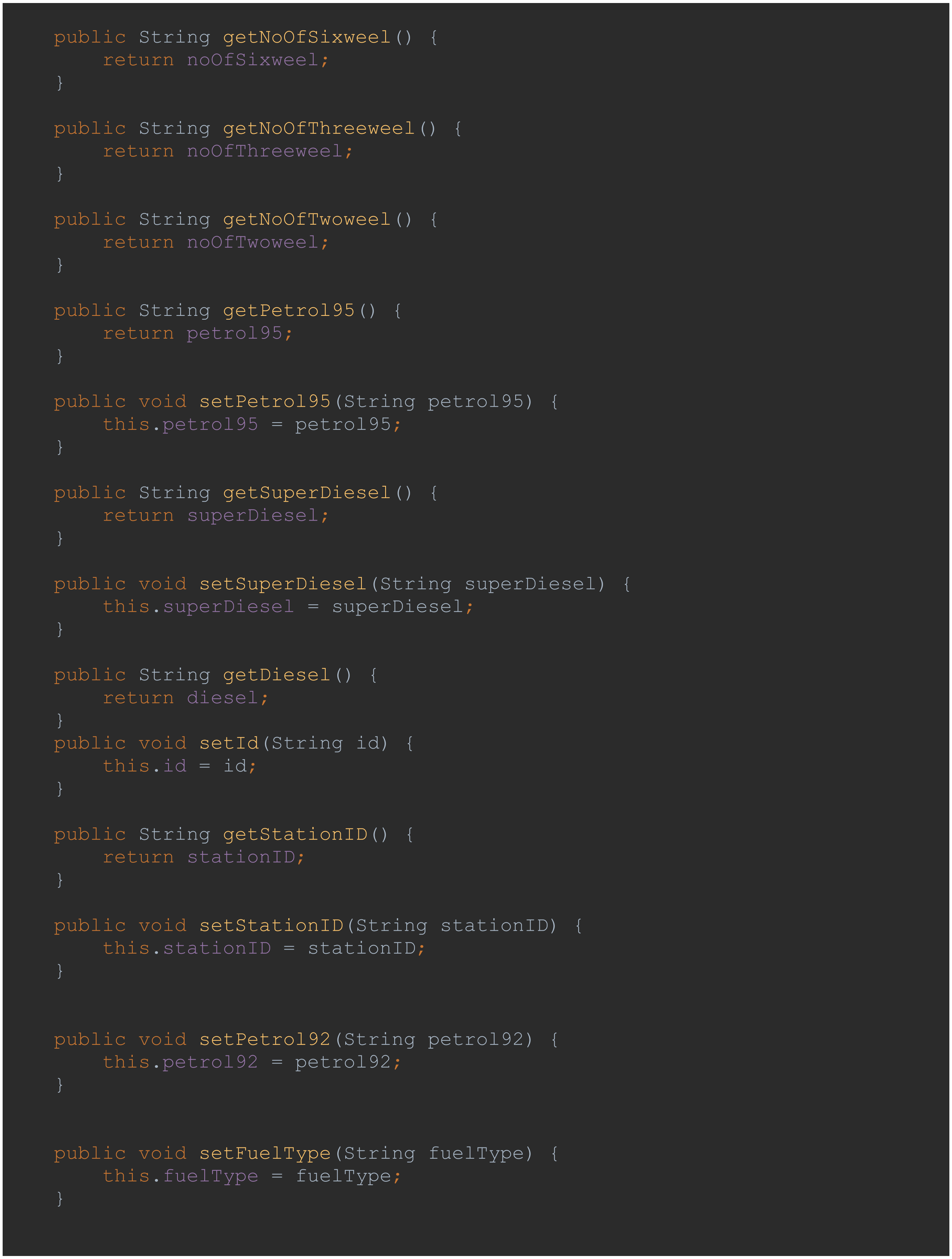
```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class FueltypeMainActivity extends AppCompatActivity {
    Button viewall,addstock,addcount;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fueltype_main);

        viewall = findViewById(R.id.fueltype_viewall);
        addstock = findViewById(R.id.fueltype_addfuel);
        addcount = findViewById(R.id.daily_count);

        viewall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(FueltypeMainActivity.this,
ViewallActivity.class));
            }
        });
        addstock.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(FueltypeMainActivity.this,
AddstockActivity.class));
            }
        });

        addcount.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(FueltypeMainActivity.this,
AdddailyvehiclecountActivity.class));
            }
        });
    }


}
```

FuelTypeModel.class

```
package com.example.fuel_app;

public class FuelTypeModel {


    private String id;
    private String stationID;
    private String petrol92;
    private String  fuelType;
    private String arrivalTime;
    private String finishTime;
    private String noOfFourweel;
    private String noOfSixweel;
    private String noOfThreeweel;
    private String noOfTwoweel;
    private String petrol95;
    private String superDiesel;
    private String diesel;


    public void setDiesel(String diesel) {
        this.diesel = diesel;
    }

    public FuelTypeModel(String petrol95, String superDiesel, String diesel,
String id, String stationID, String petrol92, String fuelType, String
arrivalTime, String finishTime, String noOfFourweel, String noOfSixweel,
String noOfThreeweel, String noOfTwoweel) {
        this.id = id;
        this.stationID = stationID;
        this.petrol92 = petrol92;
        this.fuelType = fuelType;
```



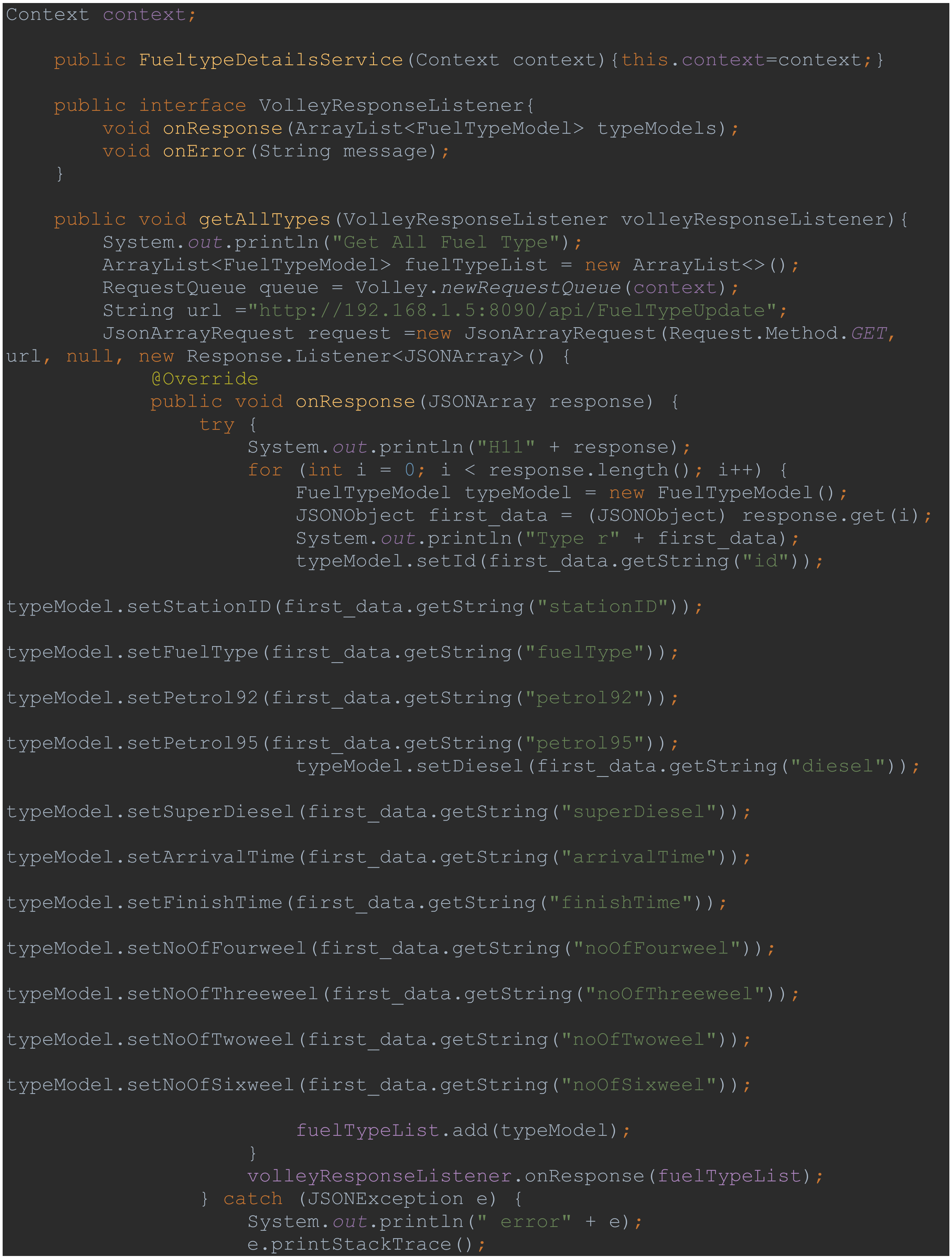
```
        this.arrivalTime = arrivalTime;
        this.finishTime = finishTime;
        this.noOfFourweel = noOfFourweel;
        this.noOfSixweel = noOfSixweel;
        this.noOfThreeweel = noOfThreeweel;
        this.noOfTwoweel = noOfTwoweel;
        this.petrol95 = petrol95;
        this.diesel = diesel;
        this.superDiesel = superDiesel;

    }
    public FuelTypeModel() {
    }

    @Override
    public String toString(){
        return "FuelTypeModel{"+
                "id='"+id+'\''+
                ", stationID='"+stationID+'\''+
                ", petrol92='"+petrol92+'\''+
                ", petrol95='"+petrol95+'\''+
                ", diesel='"+diesel+'\''+
                ", superDiesel='"+superDiesel+'\''+
                ", fuelType='"+fuelType+'\''+
                ", arrivalTime='"+arrivalTime+'\''+
                ", finishTime='"+finishTime+'\''+
                ", noOfFourweel='"+noOfFourweel+'\''+
                ", noOfSixweel='"+noOfSixweel+'\''+
                ", noOfThreeweel='"+noOfThreeweel+'\''+
                ", noOfTwoweel='"+noOfTwoweel+'\''+
                '}';
    }

    public String getId() {
        return id;
    }
    public String getPetrol92() {
        return petrol92;
    }

    public String getFuelType() {
        return fuelType;
    }

    public String getArrivalTime() {
        return arrivalTime;
    }

    public String getFinishTime() {
        return finishTime;
    }

    public String getNoOfFourweel() {
        return noOfFourweel;
    }
```



```
    public String getNoOfSixweel() {
        return noOfSixweel;
    }

    public String getNoOfThreeweel() {
        return noOfThreeweel;
    }

    public String getNoOfTwoweel() {
        return noOfTwoweel;
    }

    public String getPetrol95() {
        return petrol95;
    }

    public void setPetrol95(String petrol95) {
        this.petrol95 = petrol95;
    }

    public String getSuperDiesel() {
        return superDiesel;
    }

    public void setSuperDiesel(String superDiesel) {
        this.superDiesel = superDiesel;
    }

    public String getDiesel() {
        return diesel;
    }
    public void setId(String id) {
        this.id = id;
    }

    public String getStationID() {
        return stationID;
    }

    public void setStationID(String stationID) {
        this.stationID = stationID;
    }


    public void setPetrol92(String petrol92) {
        this.petrol92 = petrol92;
    }


    public void setFuelType(String fuelType) {
        this.fuelType = fuelType;
    }
    public void setArrivalTime(String arrivalTime) {
        this.arrivalTime = arrivalTime;
    }


    public void setFinishTime(String finishTime) {
        this.finishTime = finishTime;
    }


    public void setNoOfFourweel(String noOfFourweel) {
        this.noOfFourweel = noOfFourweel;
    }



    public void setNoOfSixweel(String noOfSixweel) {
        this.noOfSixweel = noOfSixweel;
    }



    public void setNoOfThreeweel(String noOfThreeweel) {
        this.noOfThreeweel = noOfThreeweel;
    }



    public void setNoOfTwoweel(String noOfTwoweel) {
        this.noOfTwoweel = noOfTwoweel;
    }
}
```

FuelTypeDetailsService.class

```
package com.example.fuel_app;

import android.content.Context;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonArrayRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class FueltypeDetailsService {
```



```
Context context;

    public FueltypeDetailsService(Context context){this.context=context;}

    public interface VolleyResponseListener{
        void onResponse(ArrayList<FuelTypeModel> typeModels);
        void onError(String message);
    }

    public void getAllTypes(VolleyResponseListener volleyResponseListener){
        System.out.println("Get All Fuel Type");
        ArrayList<FuelTypeModel> fuelTypeList = new ArrayList<>();
        RequestQueue queue = Volley.newRequestQueue(context);
        String url ="http://192.168.1.5:8090/api/FuelTypeUpdate";
        JsonArrayRequest request =new JsonArrayRequest(Request.Method.GET,
url, null, new Response.Listener<JSONArray>() {
            @Override
            public void onResponse(JSONArray response) {
                try {
                    System.out.println("H11" + response);
                    for (int i = 0; i < response.length(); i++) {
                        FuelTypeModel typeModel = new FuelTypeModel();
                        JSONObject first_data = (JSONObject) response.get(i);
                        System.out.println("Type r" + first_data);
                        typeModel.setId(first_data.getString("id"));

typeModel.setStationID(first_data.getString("stationID"));

typeModel.setFuelType(first_data.getString("fuelType"));

typeModel.setPetrol92(first_data.getString("petrol92"));

typeModel.setPetrol95(first_data.getString("petrol95"));
                        typeModel.setDiesel(first_data.getString("diesel"));

typeModel.setSuperDiesel(first_data.getString("superDiesel"));

typeModel.setArrivalTime(first_data.getString("arrivalTime"));

typeModel.setFinishTime(first_data.getString("finishTime"));

typeModel.setNoOfFourweel(first_data.getString("noOfFourweel"));

typeModel.setNoOfThreeweel(first_data.getString("noOfThreeweel"));

typeModel.setNoOfTwoweel(first_data.getString("noOfTwoweel"));

typeModel.setNoOfSixweel(first_data.getString("noOfSixweel"));

                        fuelTypeList.add(typeModel);
                    }
                    volleyResponseListener.onResponse(fuelTypeList);
                } catch (JSONException e) {
                    System.out.println(" error" + e);
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                System.out.println("Error :"+error);
                volleyResponseListener.onError("Error Fetching FuelType
Data");
            }
        });
        queue.add(request);
    }


}
```

FuelTypeBasedAdapter.class

```
package com.example.fuel_app;

import android.annotation.SuppressLint;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;

import java.util.ArrayList;

public class FuelTypeBasedAdapter extends ArrayAdapter<FuelTypeModel> {
    public FuelTypeBasedAdapter(@NonNull Context context,
ArrayList<FuelTypeModel> fuelTypeModelArrayList){

super(context,R.layout.activity_fueltype_listview,fuelTypeModelArrayList);
    }

    @SuppressLint("ResourceAsColor")
    @NonNull
    @Override
    public View getView(int position, @Nullable View convertView, @NonNull
ViewGroup parent){
        System.out.println("inside get type view");
        FuelTypeModel model = getItem(position);
        if(convertView == null){
            convertView =
LayoutInflater.from(getContext()).inflate(R.layout.activity_fueltype_listview
,parent,false);
        }
        System.out.println("station Name : " + model.getStationID());
        TextView stationName = convertView.findViewById(R.id.newfuelview);
        stationName.setText(model.getStationID()+"-"+model.getFuelType() );
        return convertView;

    }
}
```

AddstockActivity.class

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class AddstockActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_addstock);
    }
}
```

ViewAllActivity.class

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

import java.util.ArrayList;

public class ViewallActivity extends AppCompatActivity {
    ListView typeList;
    FuelTypeBasedAdapter listAdapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_viewall);

        typeList=(ListView) findViewById(R.id.newListView);
        FueltypeDetailsService fueltypeDetailsService=new
FueltypeDetailsService(ViewallActivity.this);
        fueltypeDetailsService.getAllTypes(new
FueltypeDetailsService.VolleyResponseListener() {
            @Override
            public void onResponse(ArrayList<FuelTypeModel> typeModel) {
                System.out.println("Type Data ss"+ typeModel);
                listAdapter=new
FuelTypeBasedAdapter(getApplicationContext(),typeModel);
                typeList.setAdapter(listAdapter);
                typeList.setClickable(true);
                typeList.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

                    @Override
                    public void onItemClick(AdapterView<?> adapterView, View
view, int i, long l) {
                        System.out.println("Clicked One"+i);
                        Intent intent=new
Intent(getApplicationContext(),ViewcountActivity.class);
                        intent.putExtra("ID",typeModel.get(i).getId());
                        startActivity(intent);
                    }
                });
            }

            @Override
            public void onError(String message) {
                Toast.makeText(ViewallActivity.this,message ,
Toast.LENGTH_LONG).show();

            }
        });
    }
}
```

ViewcountActivity.class

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;

public class ViewcountActivity extends AppCompatActivity {
    String id;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_viewcount);

        Intent intent = this.getIntent();
        if(intent != null){
            id = intent.getStringExtra("ID");
        }
        System.out.println("id : " + id);
    }
}
```

Jayasekara D.D.K. G (IT19220048)

Java Classes

SplashActivity.java

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class SplashActivity extends AppCompatActivity {

    Button vehicleownerbtn, stationownerbtn;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash);

        vehicleownerbtn = findViewById(R.id.vehicleowner);
        stationownerbtn = findViewById(R.id.stationowner);

        vehicleownerbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new
Intent(SplashActivity.this,LoginActivity.class);
                startActivity(intent);
            }
        });

        stationownerbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new
Intent(SplashActivity.this,OwnerLoginActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

RegisterActivity.java

```
package com.example.fuel_app;

        import androidx.appcompat.app.AppCompatActivity;

        import android.content.Intent;
        import android.os.Bundle;
        import android.view.View;
        import android.widget.Button;
        import android.widget.EditText;
        import android.widget.TextView;
        import android.widget.Toast;

        import com.android.volley.RequestQueue;
        import com.android.volley.Response;
        import com.android.volley.VolleyError;
        import com.android.volley.VolleyLog;
        import com.android.volley.toolbox.JsonObjectRequest;
        import com.android.volley.toolbox.Volley;

        import org.json.JSONException;
        import org.json.JSONObject;

        import java.util.HashMap;

public class RegisterActivity extends AppCompatActivity {

    Button signinbtn;
    EditText vehiclenum,password,vtype,ftype,chesis;
    private DBHelper DB;

    private RequestQueue requestQueue;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        DB = new DBHelper(this);
        signinbtn = findViewById(R.id.userregibtn);
        vehiclenum = findViewById(R.id.vehiclenumber);
        password = findViewById(R.id.password);
        vtype = findViewById(R.id.vehicletype);
        ftype = findViewById(R.id.fueltype);
        chesis = findViewById(R.id.chesisnumber);

        signinbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Intent intent = new
Intent(RegisterActivity.this,LoginActivity.class);
               // startActivity(intent);

                String vnum,pw,v_type,f_type,chesis_n;
                vnum = vehiclenum.getText().toString();
                pw = password.getText().toString();
                v_type = vtype.getText().toString();
                f_type = ftype.getText().toString();
                chesis_n = chesis.getText().toString();

                if(vnum.equals("")|| pw.equals("")|| v_type.equals("")||
f_type.equals("")|| chesis_n.equals("")){
                    Toast.makeText(RegisterActivity.this, "You should fill all
fields !!", Toast.LENGTH_SHORT).show();
                }
                else{
                    Boolean checkUser = DB.checkusername(vnum);
                    if(checkUser == false){
                        //postDataToDB(vnum,pw,v_type,f_type,chesis_n);
                        Boolean insertSuccess = DB.insertData(vnum, pw);
                        if(insertSuccess == true){
                            Toast.makeText(RegisterActivity.this, "User
Registered Successfully !", Toast.LENGTH_SHORT).show();
                            Intent intent1 = new Intent(RegisterActivity.this,
LoginActivity.class);
                            startActivity(intent1);
                        }else {
                            Toast.makeText(RegisterActivity.this, "Something went
wrong !", Toast.LENGTH_SHORT).show();
                        }
                    }
                    else {
                        Toast.makeText(RegisterActivity.this, "User already
registered !", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
    }
    // implementation to post the vehicle owner data to DB
    private void postDataToDB(String vnum, String pw, String v_type, String 
f_type, String chesis_n ){
        try {
            // url to post the user data
            String url = "http://192.168.1.11:8081/api/Customer";
            HashMap<String, String> params = new HashMap<String, String>();
            params.put("vehicleNb", vnum);
            params.put("password", pw);
            params.put("vehicleType", v_type);
            params.put("fuelType", f_type);
            params.put("chesisNb", chesis_n);
            JsonObjectRequest req = new JsonObjectRequest(url, new 
JSONObject(params),
                    new Response.Listener<JSONObject>() {
                        @Override
                        public void onResponse(JSONObject response) {
                            try {
                                VolleyLog.v("Response:%n %s", 
response.toString(4));
                            } catch (JSONException e) {
                                e.printStackTrace();
                            }
                        }
                    }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    VolleyLog.e("Error: ", error.getMessage());
                }
            });
            requestQueue = Volley.newRequestQueue(RegisterActivity.this);
            requestQueue.add(req);
        }catch (Exception e){
        }
    }
}
```

LoginActivity.java

```
package com.example.fuel_app;
```

```
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {

    TextView signinbtn;
    Button vehicleLoginButton;
    EditText vehiclenum,password;
    private DBHelper DB;

    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        vehicleLoginButton = findViewById(R.id.vehicleLoginBtn);
        vehiclenum = findViewById(R.id.ve_num);
        password = findViewById(R.id.passw);
        DB = new DBHelper(this);
        vehicleLoginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Intent intent = new
Intent(LoginActivity.this,Home_Screen.class);
                //startActivity(intent);

                String vnum,pw;
                vnum = vehiclenum.getText().toString();
                pw = password.getText().toString();

                if(vnum.equals("")|| pw.equals("")){
                    Toast.makeText(LoginActivity.this, "Required fields !!",
Toast.LENGTH_SHORT).show();
                }
                else{
                    Boolean checkUser = DB.checkusername(vnum);
                    if(checkUser == false){
                        //postDataToDB(vnum,pw);
                        Boolean insertSuccess = DB.insertData(vnum, pw);
                        if(insertSuccess == true){
                            Toast.makeText(LoginActivity.this, "User login
Successfully !", Toast.LENGTH_SHORT).show();
                            Intent intent1 = new Intent(LoginActivity.this,
Home_Screen.class);
                            startActivity(intent1);
                        }else {
                            Toast.makeText(LoginActivity.this, "Something went
wrong !", Toast.LENGTH_SHORT).show();
                        }
                    }
                }
            }
        });

        signinbtn = findViewById(R.id.signup);

        signinbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new
Intent(LoginActivity.this,RegisterActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

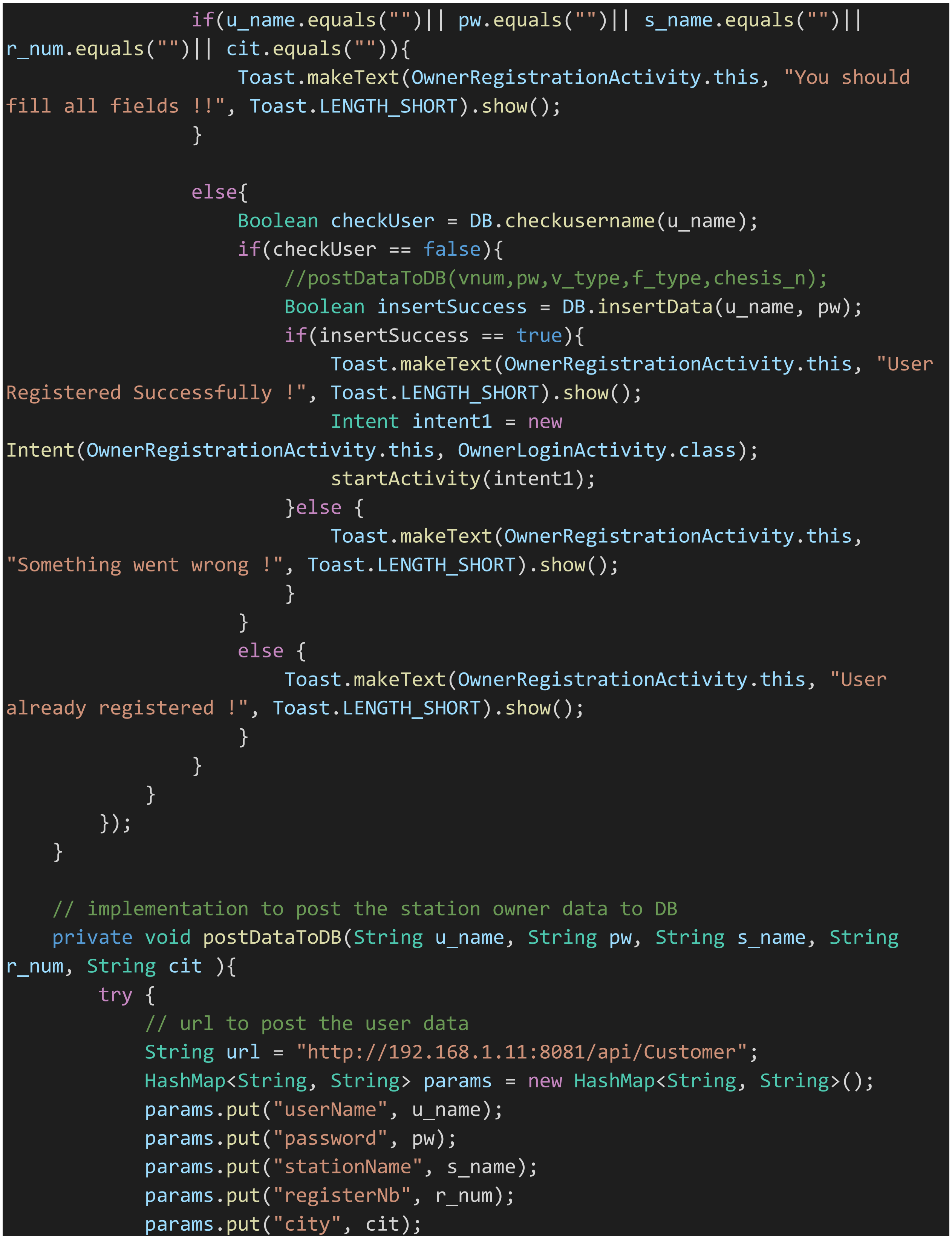
OwnerLoginActivity.java

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class OwnerLoginActivity extends AppCompatActivity {

    TextView signinbtn;
    Button login;
    EditText username,password;
    private DBHelperOwner DB;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_owner_login);

        login =findViewById(R.id.login);
        DB = new DBHelperOwner(this);
        username = findViewById(R.id.usern);
        password = findViewById(R.id.pa);

        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Intent intent = new
Intent(OwnerLoginActivity.this,FueltypeMainActivity.class);
                //startActivity(intent);

                String unum,pw;
                unum = username.getText().toString();
                pw = password.getText().toString();

                if(unum.equals("")|| pw.equals("")){
                    Toast.makeText(OwnerLoginActivity.this, "You should fill all
fields !!", Toast.LENGTH_SHORT).show();
                }

                else{
                    Boolean checkUser = DB.checkusername(unum);
                    if(checkUser == false){
                        //postDataToDB(vnum,pw,v_type,f_type,chesis_n);
                        Boolean insertSuccess = DB.insertData(unum, pw);
                        if(insertSuccess == true){
                            Toast.makeText(OwnerLoginActivity.this, "Owner Login
Successfully !", Toast.LENGTH_SHORT).show();
```

```
                            Intent intent1 = new Intent(OwnerLoginActivity.this, 
FueltypeMainActivity.class);
                            startActivity(intent1);
                        }else {
                            Toast.makeText(OwnerLoginActivity.this, "Something 
went wrong !", Toast.LENGTH_SHORT).show();
                        }
                    }

                }
            }
        });

        signinbtn = findViewById(R.id.ownersignup);

        signinbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new 
Intent(OwnerLoginActivity.this,OwnerRegistrationActivity.class);
                startActivity(intent);
            }
        });

    }
}
```

OwnerRegistrationActivity.java

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;

public class OwnerRegistrationActivity extends AppCompatActivity {

    Button ownerregi;
    EditText uname,password,sname,rnum,city;
    private DBHelperOwner DB;

    private RequestQueue requestQueue;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_owner_registration);
        DB = new DBHelperOwner(this);
        ownerregi = findViewById(R.id.ownerregibtn);
        uname = findViewById(R.id.username);
        password = findViewById(R.id.password);
        sname = findViewById(R.id.stationname);
        rnum = findViewById(R.id.registernumber);
        city = findViewById(R.id.city);

        ownerregi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Intent intent = new
Intent(OwnerRegistrationActivity.this,OwnerLoginActivity.class);
                //startActivity(intent);

                String u_name,pw,s_name,r_num,cit;
                u_name = uname.getText().toString();
                pw = password.getText().toString();
                s_name = sname.getText().toString();
                r_num = rnum.getText().toString();
                cit = city.getText().toString();
```



```
                if(u_name.equals("")|| pw.equals("")|| s_name.equals("")||
r_num.equals("")|| cit.equals("")){
                    Toast.makeText(OwnerRegistrationActivity.this, "You should
fill all fields !!", Toast.LENGTH_SHORT).show();
                }

                else{
                    Boolean checkUser = DB.checkusername(u_name);
                    if(checkUser == false){
                        //postDataToDB(vnum,pw,v_type,f_type,chesis_n);
                        Boolean insertSuccess = DB.insertData(u_name, pw);
                        if(insertSuccess == true){
                            Toast.makeText(OwnerRegistrationActivity.this, "User
Registered Successfully !", Toast.LENGTH_SHORT).show();
                            Intent intent1 = new
Intent(OwnerRegistrationActivity.this, OwnerLoginActivity.class);
                            startActivity(intent1);
                        }else {
                            Toast.makeText(OwnerRegistrationActivity.this,
"Something went wrong !", Toast.LENGTH_SHORT).show();
                        }
                    }
                    else {
                        Toast.makeText(OwnerRegistrationActivity.this, "User
already registered !", Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
    }

    // implementation to post the station owner data to DB
    private void postDataToDB(String u_name, String pw, String s_name, String
r_num, String cit ){
        try {
            // url to post the user data
            String url = "http://192.168.1.11:8081/api/Customer";
            HashMap<String, String> params = new HashMap<String, String>();
            params.put("userName", u_name);
            params.put("password", pw);
            params.put("stationName", s_name);
            params.put("registerNb", r_num);
            params.put("city", cit);
            JsonObjectRequest req = new JsonObjectRequest(url, new
JSONObject(params),
                    new Response.Listener<JSONObject>() {
                        @Override
                        public void onResponse(JSONObject response) {
                            try {
                                VolleyLog.v("Response:%n %s",
response.toString(4));
                            } catch (JSONException e) {
                                e.printStackTrace();
                            }
                        }
                    }, new Response.ErrorListener() {
                @Override
                public void onErrorResponse(VolleyError error) {
                    VolleyLog.e("Error: ", error.getMessage());
                }
            });
            requestQueue =
Volley.newRequestQueue(OwnerRegistrationActivity.this);
            requestQueue.add(req);
        }catch (Exception e){
        }
    }
}
```

UserManagementActivity.java

```
package com.example.fuel_app;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class UserManagementActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_user_management);
    }
}
```

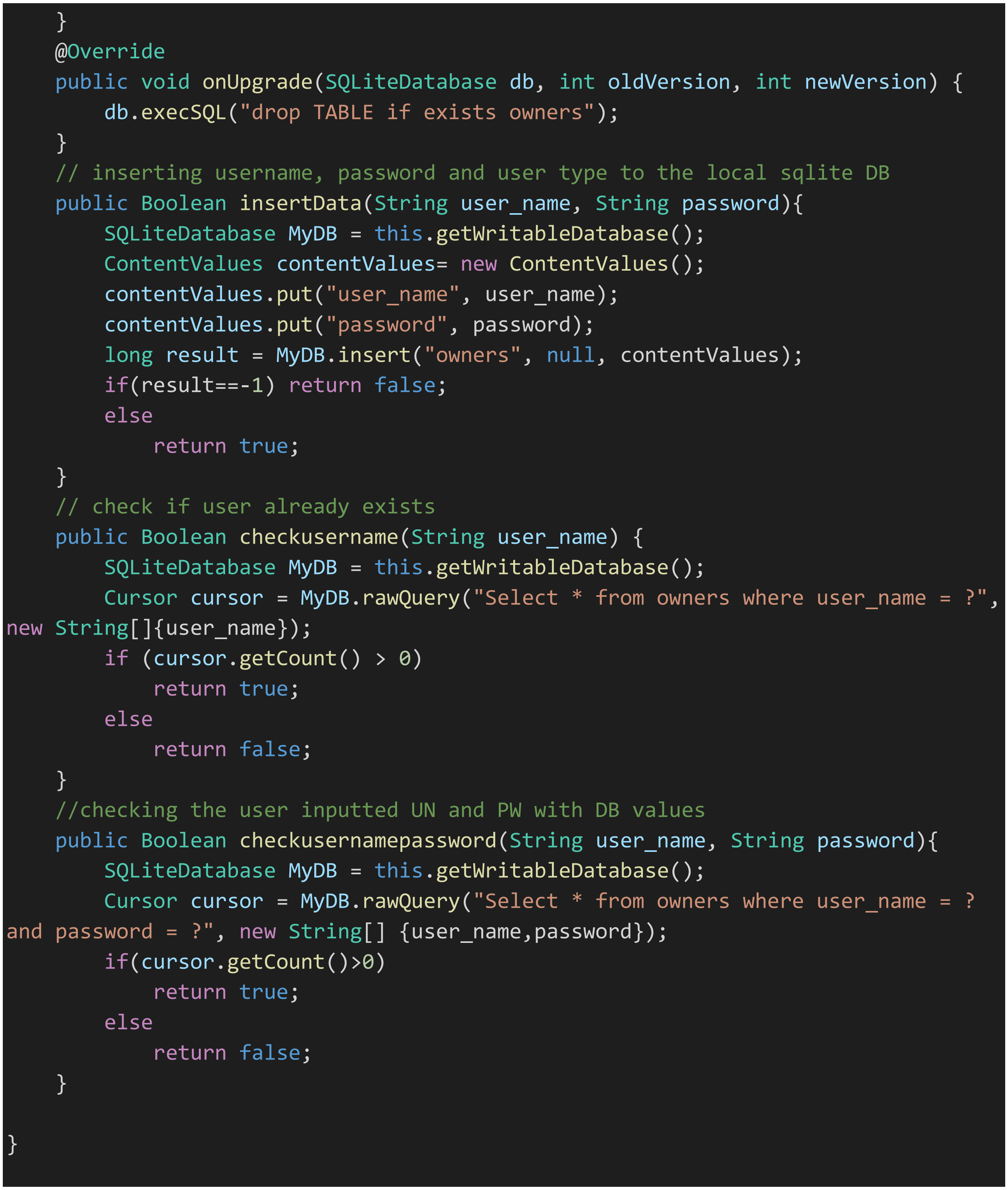
Other Classes

DBHelper.java

```
package com.example.fuel_app;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
public class DBHelper extends SQLiteOpenHelper{

    public String USERTYPE;
    // initializing the DB name is SQLite
    public static final String DBNAME = "UserLogin.db";
    private static final String TAG = "tag";
    // constructor
    public DBHelper(Context context) {
        super(context, "UserLogin.db", null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create Table users(vehicle_number TEXT primary key, password
TEXT)"); // creating the users table
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("drop TABLE if exists users");
    }
    // inserting username, password and user type to the local sqlite DB
    public Boolean insertData(String vehicle_number, String password){
        SQLiteDatabase MyDB = this.getWritableDatabase();
        ContentValues contentValues= new ContentValues();
        contentValues.put("vehicle_number", vehicle_number);
        contentValues.put("password", password);
        long result = MyDB.insert("users", null, contentValues);
        if(result==-1) return false;
        else
            return true;
    }
    // check if user already exists
    public Boolean checkusername(String vehicle_number) {
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from users where vehicle_number =
?", new String[]{vehicle_number});
        if (cursor.getCount() > 0)
            return true;
        else
            return false;
    }
    //checking the user inputted UN and PW with DB values
    public Boolean checkusernamepassword(String vehicle_number, String password){
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from users where vehicle_number =
? and password = ?", new String[] {vehicle_number,password});
        if(cursor.getCount()>0)
            return true;
        else
            return false;
    }

}
```

DBHelperOwner.java

```
package com.example.fuel_app;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DBHelperOwner extends SQLiteOpenHelper {
    public String USERTYPE;
    // initializing the DB name is SQLite
    public static final String DBNAME = "UserLogin.db";
    private static final String TAG = "tag";
    // constructor
    public DBHelperOwner(Context context) {
        super(context, "OwnerLogin.db", null, 1);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("create Table owners(user_name TEXT primary key, password
TEXT)"); // creating the users table
```



```
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("drop TABLE if exists owners");
    }
    // inserting username, password and user type to the local sqlite DB
    public Boolean insertData(String user_name, String password){
        SQLiteDatabase MyDB = this.getWritableDatabase();
        ContentValues contentValues= new ContentValues();
        contentValues.put("user_name", user_name);
        contentValues.put("password", password);
        long result = MyDB.insert("owners", null, contentValues);
        if(result==-1) return false;
        else
            return true;
    }
    // check if user already exists
    public Boolean checkusername(String user_name) {
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from owners where user_name = ?",
new String[]{user_name});
        if (cursor.getCount() > 0)
            return true;
        else
            return false;
    }
    //checking the user inputted UN and PW with DB values
    public Boolean checkusernamepassword(String user_name, String password){
        SQLiteDatabase MyDB = this.getWritableDatabase();
        Cursor cursor = MyDB.rawQuery("Select * from owners where user_name = ?
and password = ?", new String[] {user_name,password});
        if(cursor.getCount()>0)
            return true;
        else
            return false;
    }

}
```

4.2 Backend-

Wishwajith W.L (IT19140230)

Controller

FuelQueueUpdateController.cs

```
using backend.Model;
using backend.Service;
using Microsoft.AspNetCore.Mvc;
using MongoDB.Driver;

// For more information on enabling Web API for empty projects, visit
https://go.microsoft.com/fwlink/?LinkID=397860

namespace backend.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class FuelQueueUpdateController : ControllerBase
    {
        private readonly IFuelQueueUpdateService queueService;

        public FuelQueueUpdateController(IFuelQueueUpdateService updateService) {
            this.queueService = updateService;
        }

        // GET: api/<FuelQueueUpdateController>
        [HttpGet]
        public ActionResult<List<FuelQueueUpdate>> Get()
        {
            return queueService.Get();
        }

        // GET api/<FuelQueueUpdateController>/5
        [HttpGet("{id}")]
        public ActionResult<FuelQueueUpdate> Get(string id)
        {
            var queue = queueService.Get(id);
            if (queue == null)
            {
                return NotFound($"customer with id = {id} not found");
            }
            return queue;
        }

        // POST api/<FuelQueueUpdateController>
        [HttpPost]
        public ActionResult<FuelQueueUpdate> Post([FromBody] FuelQueueUpdate value)
        {
            queueService.Create(value);
            return CreatedAtAction(nameof(Get), new { id = value.Id }, value);
        }

        // PUT api/<FuelQueueUpdateController>/5
        [HttpPut("{id}")]
        public ActionResult Put(string id, [FromBody] FuelQueueUpdate value)
        {
            var existingcustomer = queueService.Get(id);
            if (existingcustomer == null)
            {
                return NotFound($"Customer with id = {id} not found");
            }
            queueService.Update(id, value);
            return NoContent();
        }

        // DELETE api/<FuelQueueUpdateController>/5
        [HttpDelete("{id}")]
        public ActionResult Delete(string id)
        {
            var queue = queueService.Get(id);
            if (queue == null)
            {
                return NotFound($"Customer with id = {id} not found");
            }
            queueService.Remove(queue.Id);
            return Ok($"Customer with id = {id} deleted");
        }
    }
}
```

Modal

FuelQueueUpdate.cs

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;

namespace backend.Model
{
    [BsonIgnoreExtraElements]
    public class FuelQueueUpdate

    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        public string Id { get; set; } = String.Empty;

        [BsonElement("checkintime")]
        public string CheckInTime { get; set; } = String.Empty;

        [BsonElement("checkouttime")]
        public string CheckOutTime { get; set; } = String.Empty;

        [BsonElement("nooflitters")]
        public string NoOfLiters { get; set; } = String.Empty;
    }
}
```

FuelQueueUpdateSettings.cs

```
namespace backend.Model
{
    public class FuelQueueUpdateSettings : IFuelQueueUpdate
    {
        public string CollectionName { get; set; } = String.Empty;
        public string CollectionString { get; set; } = String.Empty;
        public string DatabseName { get ; set ; } = String.Empty;
    }
}
```

IFuelQueueUpdateSettings.cs

```
namespace backend.Model
{
    public interface IFuelQueueUpdate
    {
        string CollectionName { get; set;  }
        string CollectionString { get; set; }
        string DatabseName { get; set; }
    }
}
```

Service

FuelQueueUpdateService.cs

```
using backend.Model;
using MongoDB.Driver;

namespace backend.Service
{
    public class FuelQueueUpdateService : IFuelQueueUpdateService
    {
        private readonly IMongoCollection<FuelQueueUpdate> _queue;

        public FuelQueueUpdateService (IFuelQueueUpdate settings, IMongoClient 
mongoClient)
        {
            var database = mongoClient.GetDatabase(settings.DatabseName);
            _queue = 
database.GetCollection<FuelQueueUpdate>(settings.CollectionName);
        }
        public FuelQueueUpdate Create(FuelQueueUpdate fuelQueue)
        {
            _queue.InsertOne(fuelQueue);
            return fuelQueue;
        }

        public List<FuelQueueUpdate> Get()
        {
            return _queue.Find(queue => true).ToList();
        }

        public FuelQueueUpdate Get(string id)
        {
            return _queue.Find(queue => queue.Id == id).FirstOrDefault();
        }

        public void Remove(string id)
        {
            _queue.DeleteOne(queue => queue.Id == id);
        }

        public void Update(string id, FuelQueueUpdate fuelQueue)
        {
            _queue.ReplaceOne(queue => queue.Id == id, fuelQueue); ;
        }
    }
}
```

IFuelQueueUpdateService.cs

```
using backend.Model;

namespace backend.Service
{
    public interface IFuelQueueUpdateService
    {
        List<FuelQueueUpdate> Get();
        FuelQueueUpdate Get(string id);
        FuelQueueUpdate Create(FuelQueueUpdate fuelQueue);
        void Update(string id,FuelQueueUpdate fuelQueue);
        void Remove(string id);

    }
}
```

Kavishka K.A.L (IT19122656)

Controller

FuelTypeUpdateControler.cs

```
using backend.Model.FuelTypeUpdates;
using backend.Service.FuelTypeUpdates;
using Microsoft.AspNetCore.Mvc;

namespace backend.Controllers
{

    [Route("api/[controller]")]
    [ApiController]
    public class FuelTypeUpdateController : ControllerBase
    {
        private readonly IFuelTypeUpdateService fuelTypeUpdateService;
        public FuelTypeUpdateController(IFuelTypeUpdateService 
fuelTypeUpdateService)
        {
            this.fuelTypeUpdateService = fuelTypeUpdateService;
        }

        // GET: api/<FuelTypeUpdate>
        [HttpGet]
        public ActionResult<List<FuelTypeUpdate>> Get()
        {
            return fuelTypeUpdateService.Get();
        }

        // GET api/<FuelTypeUpdate>/5
        [HttpGet("{id}")]
        public ActionResult<FuelTypeUpdate> Get(string id)
        {


            var fueltypeupdate = fuelTypeUpdateService.Get(id);
            if (fueltypeupdate == null)
            {
                return NotFound($"supplier with id = {id} not found");
            }
            return fueltypeupdate;
        }

        // GET api/<FuelTypeUpdate>/5
        [HttpGet("getStations/{id}")]
        public ActionResult<FuelTypeUpdate> GetStations(string id)
        {


            var fueltypeupdate = fuelTypeUpdateService.GetStations(id);
            if (fueltypeupdate == null)
            {
                return NotFound($"supplier with id = {id} not found");
            }
            return fueltypeupdate;
        }

        // POST api/<FuelTypeUpdate>
        [HttpPost]
        public ActionResult<FuelTypeUpdate> Post([FromBody] FuelTypeUpdate 
fuelTypeUpdate)
        {
            fuelTypeUpdateService.Create(fuelTypeUpdate);
            return CreatedAtAction(nameof(Get), new { id = fuelTypeUpdate.Id }, 
fuelTypeUpdate);
        }

        // PUT api/<FuelTypeUpdate>/5
        [HttpPut("{id}")]
        public ActionResult Put(string id, [FromBody] FuelTypeUpdate fuelTypeUpdate)
        {
            var existingfueltype = fuelTypeUpdateService.Get(id);
            if (existingfueltype == null)
            {
                return NotFound($"Supplier with id = {id} not found");
            }
            fuelTypeUpdateService.Update(id, fuelTypeUpdate);
            return NoContent();


        }

        // DELETE api/<FuelTypeUpdate>/5
        [HttpDelete("{id}")]
        public ActionResult Delete(String id)
        {


            var fuelTypeUpdate = fuelTypeUpdateService.Get(id);
            if (fuelTypeUpdate == null)
            {
                return NotFound($"supplier with id = {id} not found");
            }
            fuelTypeUpdateService.Remove(fuelTypeUpdate.Id);
            return Ok($"supplier with id = {id} deleted");
        }

    }
}
```

Modal

FuelTypeUpdate.cs

```
using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;


namespace backend.Model.FuelTypeUpdates
{

    [BsonIgnoreExtraElements]
    public class FuelTypeUpdate
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        public string Id { get; set; } = string.Empty;

        [BsonElement("stationID")]
        public string StationID { get; set; } = string.Empty;


        [BsonElement("arrivaltime")]
        public string ArrivalTime { get; set; } = string.Empty;

        [BsonElement("fuelType")]
        public string FuelType { get; set; } = string.Empty;

        [BsonElement("noOffourweel")]
        public string NoOfFourweel { get; set; } = string.Empty;

        [BsonElement("noOfsixweel")]
        public string NoOfSixweel { get; set; } = string.Empty;

        [BsonElement("noofthreeweel")]
        public string NoOfThreeweel { get; set; } = string.Empty;

        [BsonElement("nooftwoweel")]
        public string NoOfTwoweel { get; set; } = string.Empty;

        [BsonElement("petrol92")]
        public string petrol92 { get; set; } = string.Empty;

        [BsonElement("petrol95")]
        public string petrol95 { get; set; } = string.Empty;

        [BsonElement("diesel")]
        public string diesel { get; set; } = string.Empty;

        [BsonElement("superDiesel")]
        public string superDiesel { get; set; } = string.Empty;

        [BsonElement("finishtime")]
        public string FinishTime { get; set; } = string.Empty;

    }
}
```

FuelTypeDBSettings.cs

```
namespace backend.Model.FuelTypeUpdates
{
    public class FuelTypeDBSettings : IFuelTypeDBSettings
    {
        public string CollectionName { get; set; } = string.Empty;
        public string ConnectionString { get; set; } = string.Empty;

        public string DatabseName { get; set; } = string.Empty;


    }
}
```

IFuelTypeDBSettings.cs

```
namespace backend.Model.FuelTypeUpdates
{
    public interface IFuelTypeDBSettings
    {
        public string CollectionName { get; set; }
        public string ConnectionString { get; set; }

        public string DatabseName { get; set; }
    }
}
```

Service

FuelTypeUpdateService.cs

```
using backend.Model.FuelTypeUpdates;

using MongoDB.Driver;

namespace backend.Service.FuelTypeUpdates
{
    public class FuelTypeUpdateService:IFuelTypeUpdateService
    {
        private readonly IMongoCollection<FuelTypeUpdate> _fuelTypeUpdate;

        public FuelTypeUpdateService(IFuelTypeDBSettings settings, IMongoClient
mongoClient)

        {
            var database = mongoClient.GetDatabase(settings.DatabseName);
            _fuelTypeUpdate =
database.GetCollection<FuelTypeUpdate>(settings.CollectionName);

        }

        public FuelTypeUpdate Create(FuelTypeUpdate fuelTypeUpdate)
        {
            _fuelTypeUpdate.InsertOne(fuelTypeUpdate);
            return fuelTypeUpdate;
        }

        public List <FuelTypeUpdate> Get()
        {
            return _fuelTypeUpdate.Find(fuelTypeUpdate => true).ToList();
        }

        public FuelTypeUpdate Get(string id)
        {
            return _fuelTypeUpdate.Find(fuelTypeUpdate => fuelTypeUpdate.Id ==
id).FirstOrDefault();
        }

        public FuelTypeUpdate GetStations(string id)
        {
            return _fuelTypeUpdate.Find(fuelTypeUpdate => fuelTypeUpdate.StationID
== id).FirstOrDefault();
        }

        public void Remove(string id)
        {
            _fuelTypeUpdate.DeleteOne(FuelTypeUpdate => FuelTypeUpdate.Id == id);
        }

        public void Update(string id,FuelTypeUpdate fuelTypeUpdate)
        {
            _fuelTypeUpdate.ReplaceOne(ftype => ftype.Id == id, fuelTypeUpdate);
        }
    }
}
```

IFuelTypeUpdateService.cs

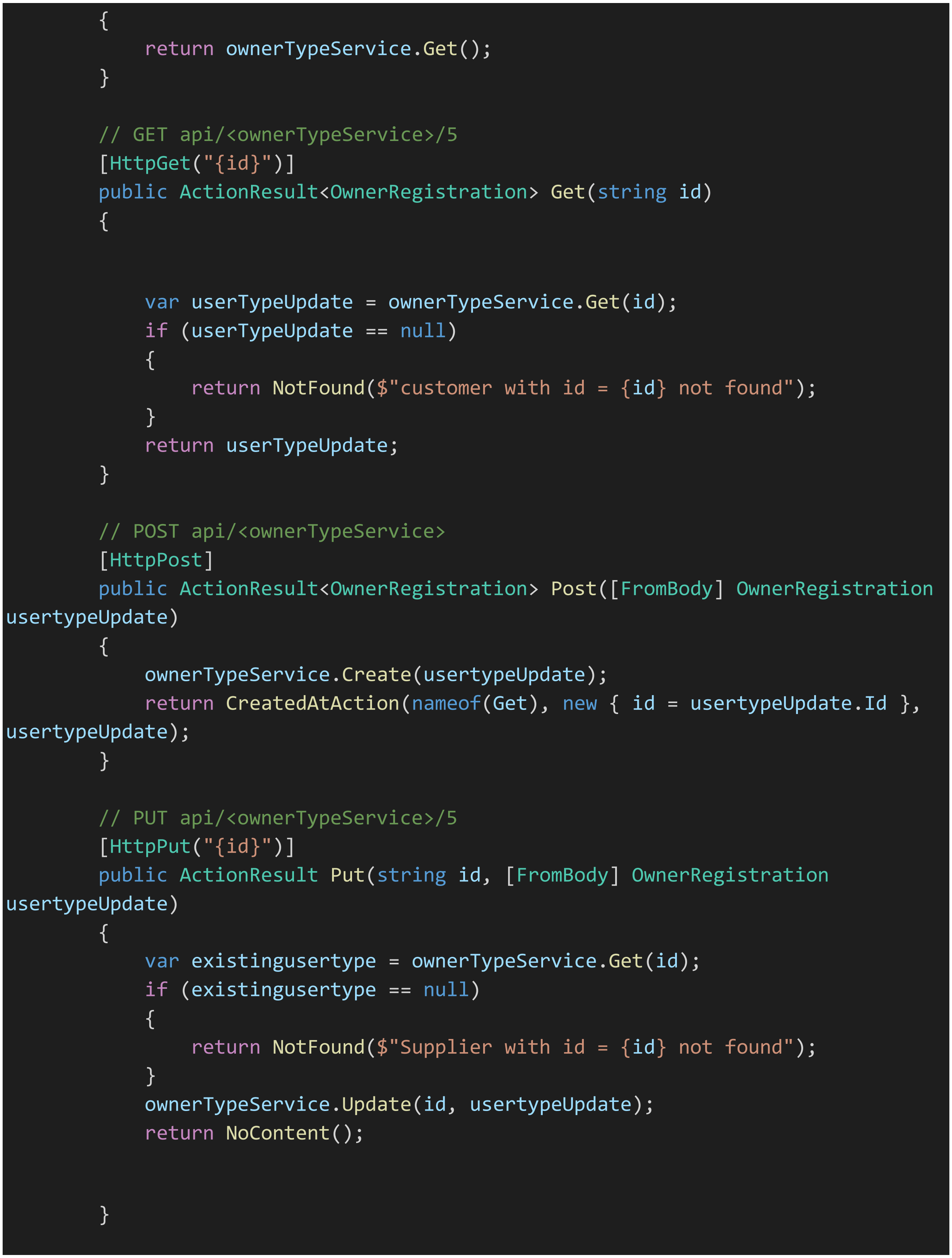
```
using backend.Model.FuelTypeUpdates;
namespace backend.Service.FuelTypeUpdates
{
    public interface IFuelTypeUpdateService
    {
        List<FuelTypeUpdate> Get();
        FuelTypeUpdate Get(string Id);
        FuelTypeUpdate GetStations(string Id);
        FuelTypeUpdate Create(
            FuelTypeUpdate fuelTypeUpdate);
        void Update(string Id, FuelTypeUpdate fuelTypeUpdate);
        void Remove(string Id);

    }
}
```

Kaveena (IT19220048)

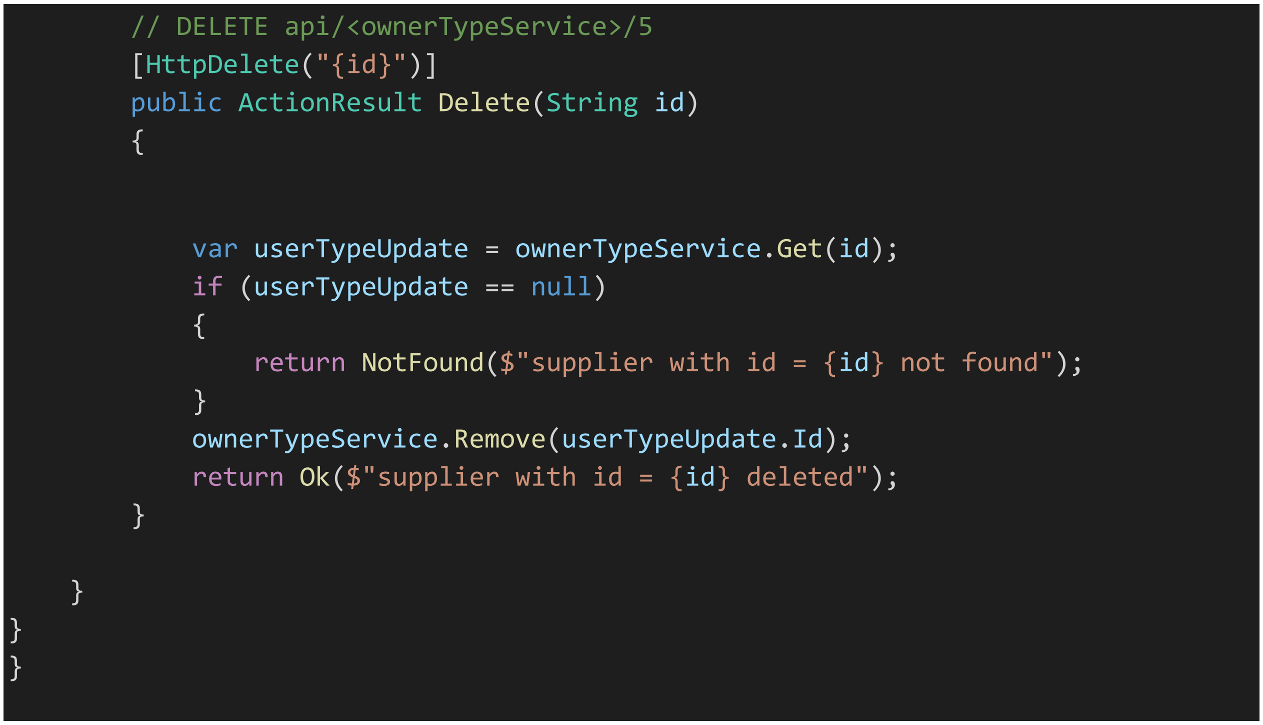
Controller –

OwnerController.cs

```
using backend.Model.OwnerLoginRegistration;
using backend.Service.OwnerService;
using Microsoft.AspNetCore.Mvc;

namespace backend.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class OwnerController : ControllerBase
    {
        private readonly IOwnerService ownerTypeService;

        public OwnerController(IOwnerService ownerService)
        {
            this.ownerTypeService = ownerService;
        }

        // GET: api/<ownerTypeService>
        [HttpGet]
        public ActionResult<List<OwnerRegistration>> Get()
```

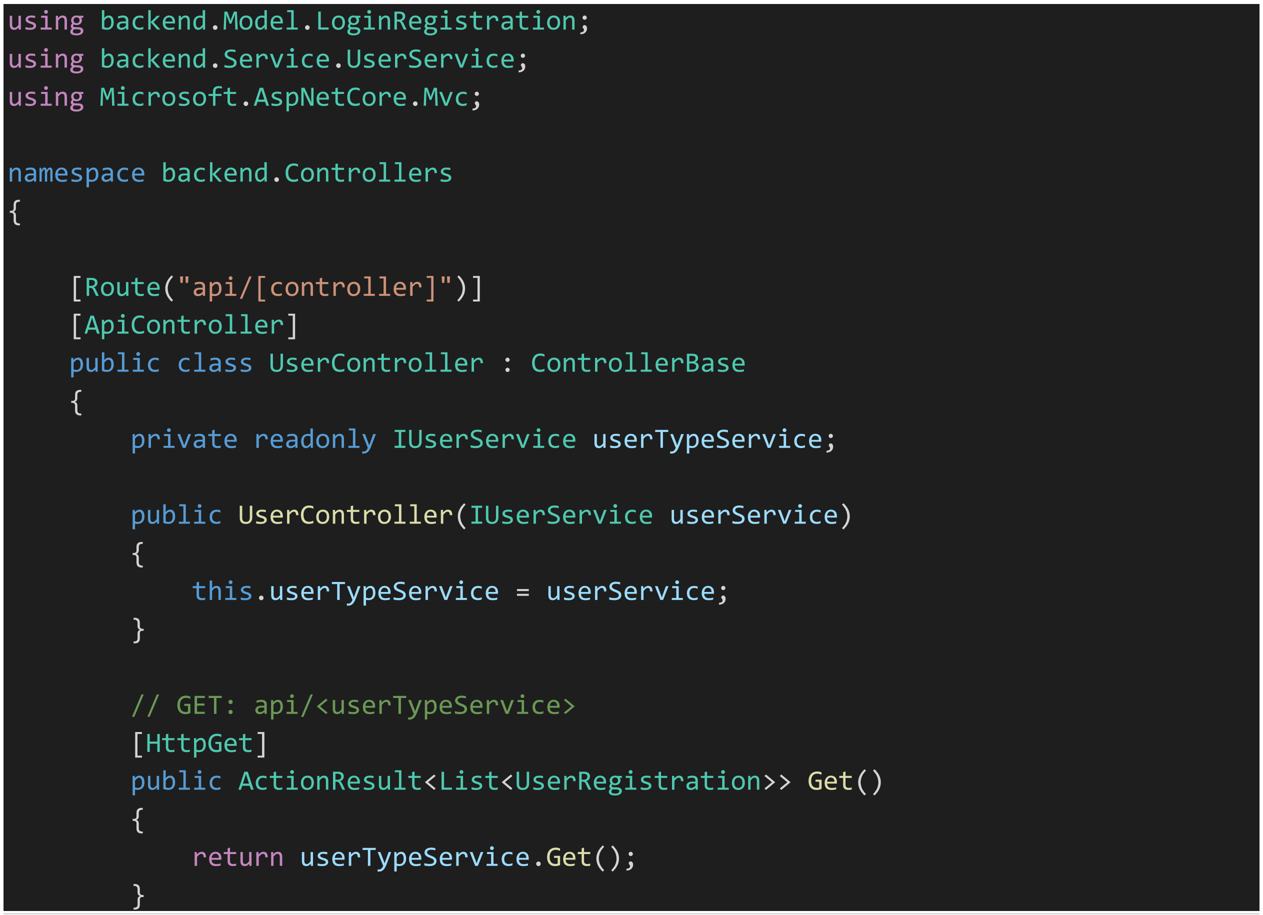


```
        {
            return ownerTypeService.Get();
        }

        // GET api/<ownerTypeService>/5
        [HttpGet("{id}")]
        public ActionResult<OwnerRegistration> Get(string id)
        {


            var userTypeUpdate = ownerTypeService.Get(id);
            if (userTypeUpdate == null)
            {
                return NotFound($"customer with id = {id} not found");
            }
            return userTypeUpdate;
        }

        // POST api/<ownerTypeService>
        [HttpPost]
        public ActionResult<OwnerRegistration> Post([FromBody] OwnerRegistration
usertypeUpdate)
        {
            ownerTypeService.Create(usertypeUpdate);
            return CreatedAtAction(nameof(Get), new { id = usertypeUpdate.Id },
usertypeUpdate);
        }

        // PUT api/<ownerTypeService>/5
        [HttpPut("{id}")]
        public ActionResult Put(string id, [FromBody] OwnerRegistration
usertypeUpdate)
        {
            var existingusertype = ownerTypeService.Get(id);
            if (existingusertype == null)
            {
                return NotFound($"Supplier with id = {id} not found");
            }
            ownerTypeService.Update(id, usertypeUpdate);
            return NoContent();


        }
```

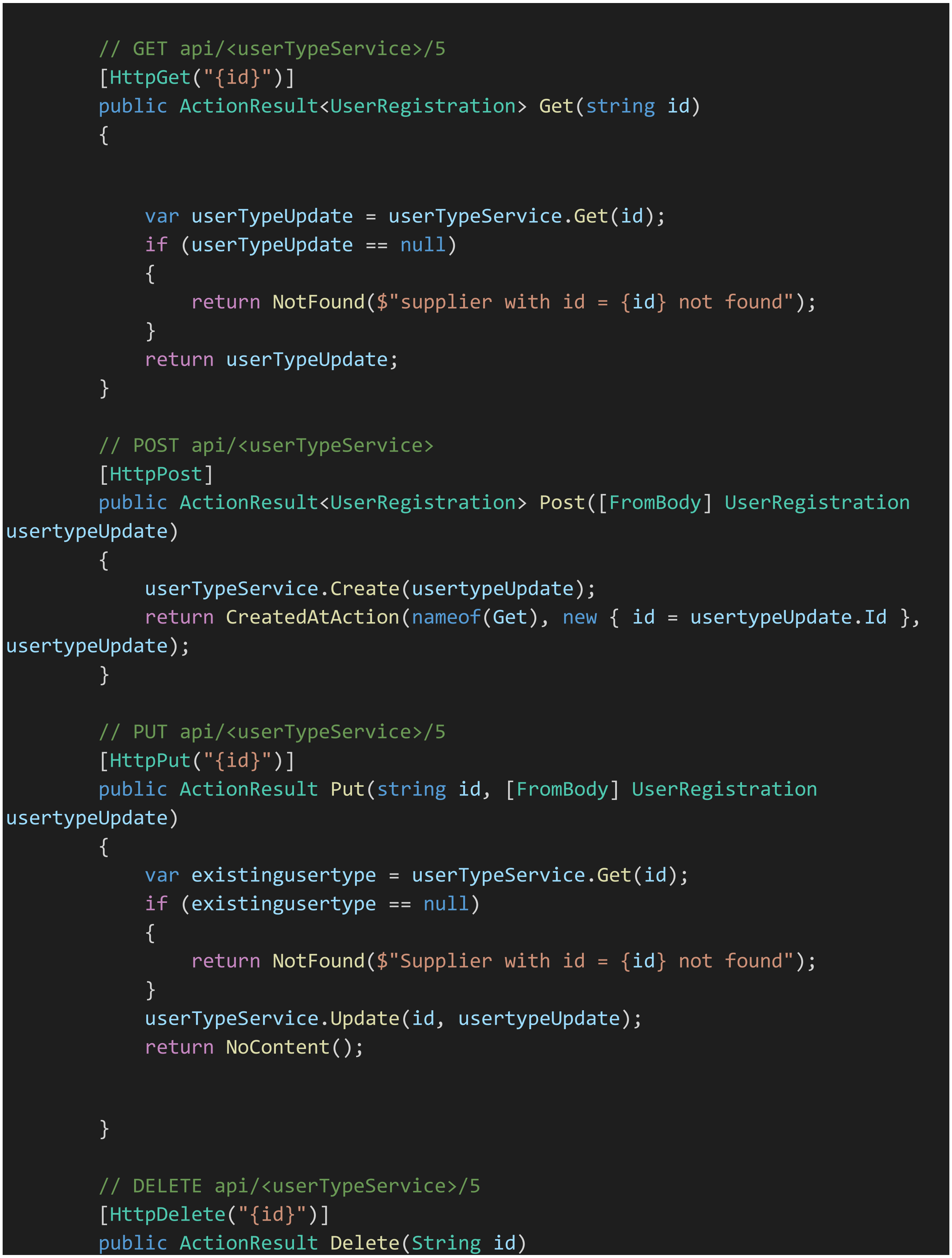


```
        // DELETE api/<ownerTypeService>/5
        [HttpDelete("{id}")]
        public ActionResult Delete(String id)
        {


            var userTypeUpdate = ownerTypeService.Get(id);
            if (userTypeUpdate == null)
            {
                return NotFound($"supplier with id = {id} not found");
            }
            ownerTypeService.Remove(userTypeUpdate.Id);
            return Ok($"supplier with id = {id} deleted");
        }

    }
}
}
```

UserController.cs



```
using backend.Model.LoginRegistration;
using backend.Service.UserService;
using Microsoft.AspNetCore.Mvc;

namespace backend.Controllers
{

    [Route("api/[controller]")]
    [ApiController]
    public class UserController : ControllerBase
    {
        private readonly IUserService userTypeService;

        public UserController(IUserService userService)
        {
            this.userTypeService = userService;
        }

        // GET: api/<userTypeService>
        [HttpGet]
        public ActionResult<List<UserRegistration>> Get()
        {
            return userTypeService.Get();
        }
```



```
        // GET api/<userTypeService>/5
        [HttpGet("{id}")]
        public ActionResult<UserRegistration> Get(string id)
        {


            var userTypeUpdate = userTypeService.Get(id);
            if (userTypeUpdate == null)
            {
                return NotFound($"supplier with id = {id} not found");
            }
            return userTypeUpdate;
        }

        // POST api/<userTypeService>
        [HttpPost]
        public ActionResult<UserRegistration> Post([FromBody] UserRegistration 
usertypeUpdate)
        {
            userTypeService.Create(usertypeUpdate);
            return CreatedAtAction(nameof(Get), new { id = usertypeUpdate.Id }, 
usertypeUpdate);
        }

        // PUT api/<userTypeService>/5
        [HttpPut("{id}")]
        public ActionResult Put(string id, [FromBody] UserRegistration 
usertypeUpdate)
        {
            var existingusertype = userTypeService.Get(id);
            if (existingusertype == null)
            {
                return NotFound($"Supplier with id = {id} not found");
            }
            userTypeService.Update(id, usertypeUpdate);
            return NoContent();


        }

        // DELETE api/<userTypeService>/5
        [HttpDelete("{id}")]
        public ActionResult Delete(String id)
        {


            var userTypeUpdate = userTypeService.Get(id);
            if (userTypeUpdate == null)
            {
                return NotFound($"supplier with id = {id} not found");
            }
            userTypeService.Remove(userTypeUpdate.Id);
            return Ok($"supplier with id = {id} deleted");
        }

    }
}
```

Modal

ILoginDBSettings.cs

```
namespace backend.Model.UserRegistration
{
    public interface ILoginDBSettings
    {
        public string CollectionName { get; set; }
        public string ConnectionString { get; set; }

        public string DatabseName { get; set; }
    }
}
```

LoginDBSettings.cs

```
namespace backend.Model.UserRegistration
{
    public class LoginDBSettings : ILoginDBSettings
    {
        public string CollectionName { get; set; } = string.Empty;
        public string ConnectionString { get; set; } = string.Empty;

        public string DatabseName { get; set; } = string.Empty;


    }
}
```

UserRegistration.cs

```

using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;

namespace backend.Model.LoginRegistration
{

    [BsonIgnoreExtraElements]
    public class UserRegistration
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        public string Id { get; set; } = string.Empty;

        [BsonElement("vehicleNb")]
        public string VehicleNb { get; set; } = string.Empty;

        [BsonElement("password")]
        public string Password { get; set; } = string.Empty;

        [BsonElement("vehicleType")]
        public string VehicleType { get; set; } = string.Empty;

        [BsonElement("fuelType")]
        public string FuelType { get; set; } = string.Empty;

        [BsonElement("chesisNb")]
        public string ChesisNb { get; set; } = string.Empty;
    }
}
```

IOwnerLoginDBSettings.cs

```
namespace backend.Model.OwnerRegistration
{
    public interface IOwnerLoginDBSettings
    {
        public string CollectionName { get; set; }
        public string ConnectionString { get; set; }

        public string DatabseName { get; set; }
    }
}
```

OwnerLoginDBSettings.cs

```
namespace backend.Model.OwnerRegistration
{
    public class OwnerLoginDBSettings : IOwnerLoginDBSettings
    {
        public string CollectionName { get; set; } = string.Empty;
        public string ConnectionString { get; set; } = string.Empty;

        public string DatabseName { get; set; } = string.Empty;


    }
}
```

OwnerRegistration.cs

```

using MongoDB.Bson.Serialization.Attributes;
using MongoDB.Bson;

namespace backend.Model.OwnerLoginRegistration
{

    [BsonIgnoreExtraElements]
    public class OwnerRegistration
    {
        [BsonId]
        [BsonRepresentation(BsonType.ObjectId)]
        public string Id { get; set; } = string.Empty;

        [BsonElement("userName")]
        public string UserName { get; set; } = string.Empty;

        [BsonElement("password")]
        public string Password { get; set; } = string.Empty;

        [BsonElement("stationName")]
        public string StationName { get; set; } = string.Empty;

        [BsonElement("registerNb")]
        public string RegisterNb { get; set; } = string.Empty;

        [BsonElement("city")]
        public string City { get; set; } = string.Empty;
    }
}
```

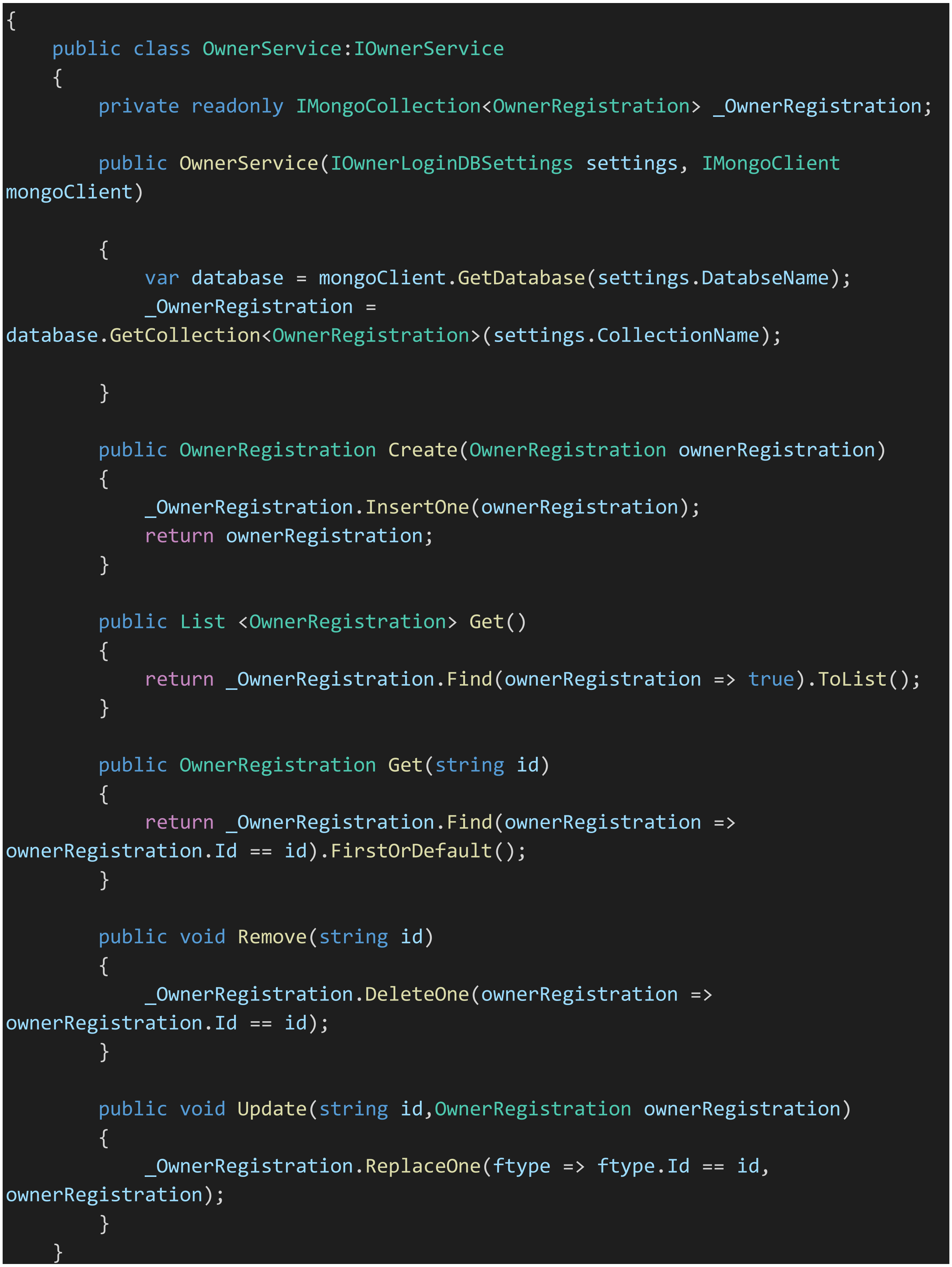
Service

IOwnerService.cs

```
using backend.Model.OwnerLoginRegistration;
using backend.Model.UserRegistration;
namespace backend.Service.OwnerService
{
    public interface IOwnerService
    {

        List<OwnerRegistration> Get();
        OwnerRegistration Get(string Id);
        OwnerRegistration Create(
            OwnerRegistration ownerRegistration);
        void Update(string Id, OwnerRegistration ownerRegistration);
        void Remove(string Id);

    }
}
```

OwnerService.cs

```
using backend.Model.OwnerLoginRegistration;
using backend.Model.OwnerRegistration;

using MongoDB.Driver;

namespace backend.Service.OwnerService
```



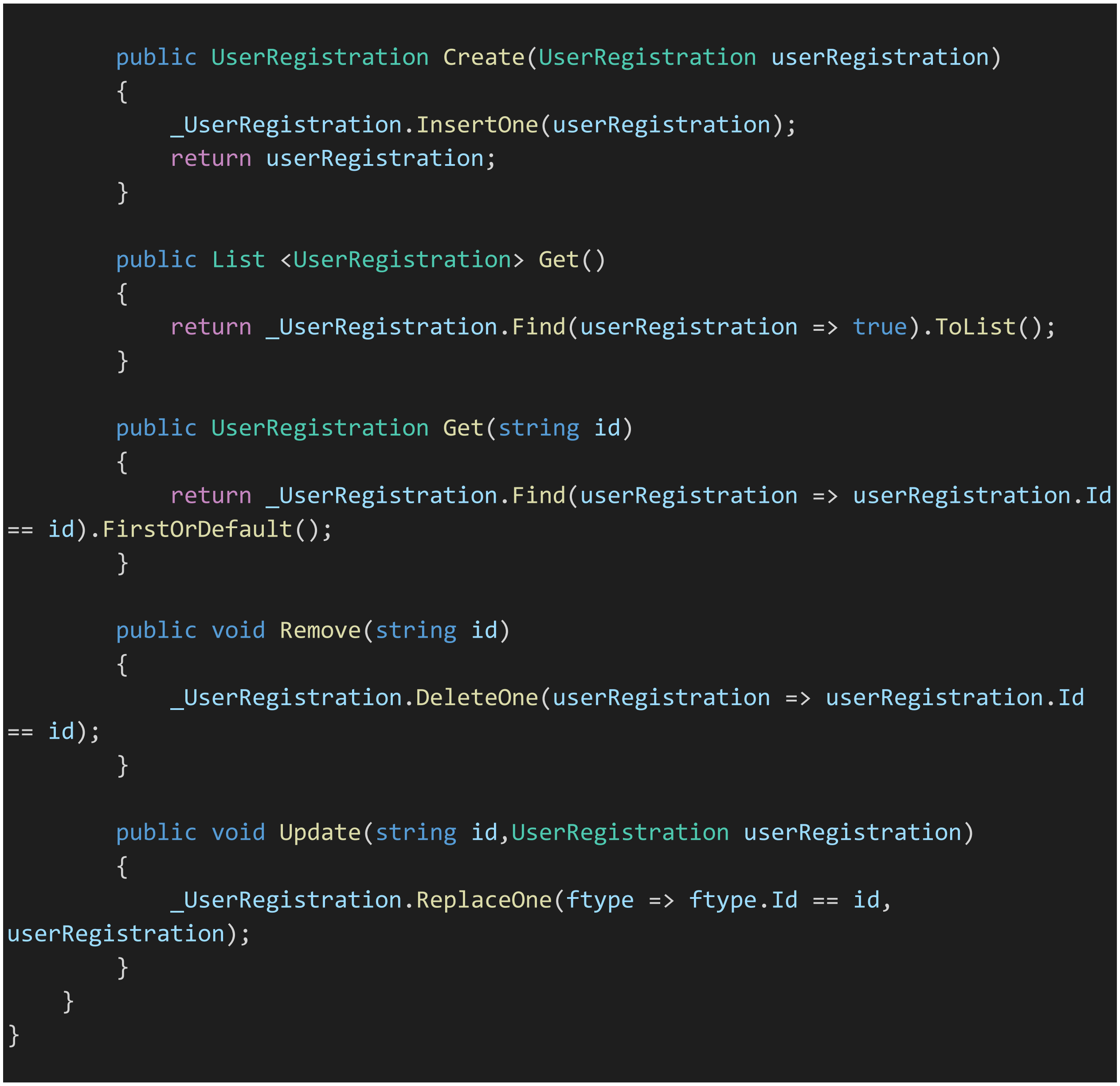
```
{
    public class OwnerService:IOwnerService
    {
        private readonly IMongoCollection<OwnerRegistration> _OwnerRegistration;

        public OwnerService(IOwnerLoginDBSettings settings, IMongoClient
mongoClient)

        {
            var database = mongoClient.GetDatabase(settings.DatabseName);
            _OwnerRegistration =
database.GetCollection<OwnerRegistration>(settings.CollectionName);

        }

        public OwnerRegistration Create(OwnerRegistration ownerRegistration)
        {
            _OwnerRegistration.InsertOne(ownerRegistration);
            return ownerRegistration;
        }

        public List <OwnerRegistration> Get()
        {
            return _OwnerRegistration.Find(ownerRegistration => true).ToList();
        }

        public OwnerRegistration Get(string id)
        {
            return _OwnerRegistration.Find(ownerRegistration =>
ownerRegistration.Id == id).FirstOrDefault();
        }

        public void Remove(string id)
        {
            _OwnerRegistration.DeleteOne(ownerRegistration =>
ownerRegistration.Id == id);
        }

        public void Update(string id,OwnerRegistration ownerRegistration)
        {
            _OwnerRegistration.ReplaceOne(ftype => ftype.Id == id,
ownerRegistration);
        }
    }
}
```

IUserService.cs

```
using backend.Model.LoginRegistration;
using backend.Model.UserRegistration;
namespace backend.Service.UserService
{
    public interface IUserService
    {

        List<UserRegistration> Get();
        UserRegistration Get(string Id);
        UserRegistration Create(
            UserRegistration userRegistration);
        void Update(string Id, UserRegistration userRegistration);
        void Remove(string Id);

    }
}
```

UserService.cs

```
using backend.Model.LoginRegistration;
using backend.Model.UserRegistration;

using MongoDB.Driver;

namespace backend.Service.UserService
{
    public class UserService:IUserService
    {
        private readonly IMongoCollection<UserRegistration> _UserRegistration;

        public UserService(ILoginDBSettings settings, IMongoClient mongoClient)

        {
            var database = mongoClient.GetDatabase(settings.DatabseName);
            _UserRegistration =
database.GetCollection<UserRegistration>(settings.CollectionName);

        }
```



```
        public UserRegistration Create(UserRegistration userRegistration)
        {
            _UserRegistration.InsertOne(userRegistration);
            return userRegistration;
        }

        public List <UserRegistration> Get()
        {
            return _UserRegistration.Find(userRegistration => true).ToList();
        }

        public UserRegistration Get(string id)
        {
            return _UserRegistration.Find(userRegistration => userRegistration.Id
== id).FirstOrDefault();
        }

        public void Remove(string id)
        {
            _UserRegistration.DeleteOne(userRegistration => userRegistration.Id
== id);
        }

        public void Update(string id,UserRegistration userRegistration)
        {
            _UserRegistration.ReplaceOne(ftype => ftype.Id == id,
userRegistration);
        }
    }
}
```

5.Individual Contribution

IT Number	Contribution
IT19140230	• View all station details list view • Display station details and fuel availability • Add check in time and checkout time with no of liters
IT19122656	• Fuel owner landing page • Add stock, add vehicle count UI and backend • View Stock UI and Backend
IT19220048	• Splash Screen • Login, Signup with User management function Frontend and Backend
Report	• All the three members are contributed equally.

6.References

Android Documentation

https://developer.android.com/docs

C# documentation

https://learn.microsoft.com/en-us/dotnet/csharp/

Get idea about android studio
https://www.youtube.com/watch?v=EknEIzswvC0&list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E

Get idea about crud tutorial
https://www.youtube.com/watch?v=L4m6fQ-IX4U
https://www.youtube.com/watch?v=8obgNNlj3Eo http://arnab.ch/blog/2013/08/asynchronous-http-requests-in-android-using-volley/

Link pages in android studio
https://www.youtube.com/watch?v=dm-jan0YORg

Get idea about c# tutorial
https://www.youtube.com/watch?v=67oWw9TanOk&list=PL82C6-O4XrHfoN_Y4MwGvJz5BntiL0z0D

Create web api using c#
https://www.tutorialsteacher.com/webapi/create-web-api-for-crud-operation