MOBILE APPLICATION DEVELOPMENT

PROJECT REPORT

DEPARTMENT OF COMPUTER SCIENCE

BAHRIA UNIVERSITY, KARACHI, PAKISTAN

PROJECT NAME

SAWAARI

GROUP MEMBERS:

1. Sohaib Aijaz (BSCS – 6A)
    - Maps
    - Select A Ride
    - Booking
    - Search Places
    - User Rides
    - Confirm Booking
    - Back End (30%)

2.Maaz Sabahuddin (BSCS – 6A)

- Login
- Signup
- Verify
- Maps
- Booking
- Reset Phone Number
- Forget Password
- Back End (70%)

INTRODUCTION:

Online bus booking system is mobile based application, that works within centralized network. It provides facility to reserve seats, cancellation of seats, and different types of inquiry which need an instant and quick reservation with multiple figures. E-ticket is the easiest and the best way to take it out.

PROBLEM:

Pakistan is served by the worst transport system. There is no transport available, even in the biggest or most working cities of Pakistan. Traveling by road in Pakistan is the most popular mode of transport as it has well- developed road network system. The express bus operators have benefited from this well-developed road network system. This industry has grown from its modest operation to providing e-ticketing system by few bus operators operating individually.

SOLUTION:

Faster bus ticketing booking and a choice of purchasing from different express bus operators, support services which are more responsive to user's needs, greater customer appreciation (through a Decision Support System), elimination of illegal bus operators and also to provide greater benefits to bus operators to enhance their business processes (through Management Information System).

REQUIREMENTS ANALYSIS:

- A customer can create account by entering his/her email or contact number, then enter a verification code and set pin to login.
- A customer can do multiple bookings.
- A customer can cancel his/her ride.
- A customer can have multiple bookings.
- A customer can modify date, time, seats of his/her ride.
- A captain can be vendor or a driver.
- A vendor can have one or more vehicles.
- A vendor can ride his/her own vehicle.
- A vendor can have multiple drivers.
- A vendor can also be a driver
- A driver can drive multiple vehicles of a same vendor.
- A driver can have only one vendor.
- A vehicle belongs to only one vendor.

CODE:

Front End on Java (Android):

Bus Activity:

```
package com.sohaibaijaz.sawaari;

import androidx.appcompat.app.AppCompatActivity;

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

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

import java.util.ArrayList;
import java.util.HashMap;

public class BusActivity extends AppCompatActivity {


    ArrayList<HashMap> buses = new ArrayList<HashMap>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bus);
        getSupportActionBar().hide();

        Button btn_back = findViewById(R.id.btn_back);
        ListView list_buses = (findViewById(R.id.list_buses));
        Bundle b = getIntent().getExtras();
        String rides_data = b.getString("rides");


        try {

            JSONArray rides = new JSONArray(rides_data);
            for(int i=0 ; i< rides.length(); i++){
                JSONObject ride = rides.getJSONObject(i);
                JSONArray ride_pickup = ride.getJSONArray("pick-up-location");
                JSONArray ride_dropoff = ride.getJSONArray("drop-off-location");


                if (ride_pickup.length()> ride_dropoff.length())
                {
                    int length_pickup_location = ride_pickup.length();

                    for(int j = 0; j<length_pickup_location; j++){

                        for(int k =0; k<ride_dropoff.length(); k++){
                            HashMap<String, String> bus = new HashMap<>();

                            JSONObject ride_pickup_object =
ride_pickup.getJSONObject(j);
                            JSONObject ride_dropoff_object =
ride_dropoff.getJSONObject(k);

                            bus.put("ride_date",
ride_pickup_object.getString("date"));
                            bus.put("vehicle_no_plate",
ride.getString("vehicle_no_plate"));
                            bus.put("seats_left", ride.getString("seats_left"));
                            bus.put("dropoff_location_id",
```

```
ride_dropoff_object.getString("stop_id"));

bus.put("dropoff_location",ride_dropoff_object.getString("stop_name"));
                            bus.put("dropoff_location_time",
ride_dropoff_object.getString("duration"));
                            bus.put("pickup_location_id",
ride_pickup_object.getString("stop_id"));
                            bus.put("pickup_location",
ride_pickup_object.getString("stop_name"));
                            bus.put("pickup_location_time",
ride_pickup_object.getString("duration"));
                            bus.put("arrival_time",
ride_pickup_object.getString("arrival_time"));
                            bus.put("pickup_distance",
ride_pickup_object.getString("distance"));
                            bus.put("dropoff_distance",
ride_dropoff_object.getString("distance"));

                            buses.add(bus);
                        }

                    }
                }

                else if (ride_dropoff.length()> ride_pickup.length())
                {
                    int length_dropoff_location = ride_dropoff.length();

                    for(int j = 0; j<length_dropoff_location; j++){

                        for(int k =0; k<ride_pickup.length(); k++){
                            HashMap<String, String> bus = new HashMap<>();

                            JSONObject ride_pickup_object =
ride_pickup.getJSONObject(k);
                            JSONObject ride_dropoff_object =
ride_dropoff.getJSONObject(j);


                            bus.put("ride_date",
ride_pickup_object.getString("date"));
                            bus.put("vehicle_no_plate",
ride.getString("vehicle_no_plate"));
                            bus.put("seats_left", ride.getString("seats_left"));
                            bus.put("dropoff_location_id",
ride_dropoff_object.getString("stop_id"));

bus.put("dropoff_location",ride_dropoff_object.getString("stop_name"));
                            bus.put("dropoff_location_time",
ride_dropoff_object.getString("duration"));
                            bus.put("pickup_location_id",
ride_pickup_object.getString("stop_id"));
                            bus.put("pickup_location",
ride_pickup_object.getString("stop_name"));
                            bus.put("pickup_location_time",
ride_pickup_object.getString("duration"));
                            bus.put("arrival_time",
ride_pickup_object.getString("arrival_time"));
                            bus.put("pickup_distance",
ride_pickup_object.getString("distance"));
                            bus.put("dropoff_distance",
ride_dropoff_object.getString("distance"));

                           buses.add(bus);
                        }

                    }
                }
```

```
                else if (ride_pickup.length() == ride_dropoff.length()){

                    int length_dropoff_location = ride_dropoff.length();
                    for(int j = 0; j<length_dropoff_location; j++){

                        for(int k =0; k<ride_pickup.length(); k++){
                            HashMap<String, String> bus = new HashMap<>();

                            JSONObject ride_pickup_object =
ride_pickup.getJSONObject(k);
                            JSONObject ride_dropoff_object =
ride_dropoff.getJSONObject(j);

                            bus.put("ride_date",
ride_pickup_object.getString("date"));
                            bus.put("vehicle_no_plate",
ride.getString("vehicle_no_plate"));
                            bus.put("seats_left", ride.getString("seats_left"));
                            bus.put("dropoff_location_id",
ride_dropoff_object.getString("stop_id"));

bus.put("dropoff_location",ride_dropoff_object.getString("stop_name"));
                            bus.put("dropoff_location_time",
ride_dropoff_object.getString("duration"));
                            bus.put("pickup_location_id",
ride_pickup_object.getString("stop_id"));
                            bus.put("pickup_location",
ride_pickup_object.getString("stop_name"));
                            bus.put("pickup_location_time",
ride_pickup_object.getString("duration"));
                            bus.put("arrival_time",
ride_pickup_object.getString("arrival_time"));
                            bus.put("pickup_distance",
ride_pickup_object.getString("distance"));
                            bus.put("dropoff_distance",
ride_dropoff_object.getString("distance"));

                              buses.add(bus);
                            }

                        }
                    }
                }



                list_buses.setAdapter(new CustomAdapterActivity(this, buses));
            }
            catch (Exception e){
                e.printStackTrace();
            }

            btn_back.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    onBackPressed();
                }
            });

        }

}
```

Custom Adapter Activity:

```
package com.sohaibaijaz.sawaari;

import androidx.appcompat.app.AppCompatActivity;

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

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

import java.util.ArrayList;
import java.util.HashMap;

public class BusActivity extends AppCompatActivity {


    ArrayList<HashMap> buses = new ArrayList<HashMap>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bus);
        getSupportActionBar().hide();

        Button btn_back = findViewById(R.id.btn_back);
        ListView list_buses = (findViewById(R.id.list_buses));
        Bundle b = getIntent().getExtras();
        String rides_data = b.getString("rides");


        try {

            JSONArray rides = new JSONArray(rides_data);
            for(int i=0 ; i< rides.length(); i++){
                JSONObject ride = rides.getJSONObject(i);
                JSONArray ride_pickup = ride.getJSONArray("pick-up-location");
                JSONArray ride_dropoff = ride.getJSONArray("drop-off-location");


                if (ride_pickup.length()> ride_dropoff.length())
                {
                    int length_pickup_location = ride_pickup.length();

                    for(int j = 0; j<length_pickup_location; j++){

                        for(int k =0; k<ride_dropoff.length(); k++){
                            HashMap<String, String> bus = new HashMap<>();

                            JSONObject ride_pickup_object =
ride_pickup.getJSONObject(j);
                            JSONObject ride_dropoff_object =
ride_dropoff.getJSONObject(k);

                            bus.put("ride_date",
ride_pickup_object.getString("date"));
                            bus.put("vehicle_no_plate",
ride.getString("vehicle_no_plate"));
                            bus.put("seats_left", ride.getString("seats_left"));
                            bus.put("dropoff_location_id",
ride_dropoff_object.getString("stop_id"));

bus.put("dropoff_location",ride_dropoff_object.getString("stop_name"));
                            bus.put("dropoff_location_time",
ride_dropoff_object.getString("duration"));
```

```
                            bus.put("pickup_location_id",
ride_pickup_object.getString("stop_id"));
                            bus.put("pickup_location",
ride_pickup_object.getString("stop_name"));
                            bus.put("pickup_location_time",
ride_pickup_object.getString("duration"));
                            bus.put("arrival_time",
ride_pickup_object.getString("arrival_time"));
                            bus.put("pickup_distance",
ride_pickup_object.getString("distance"));
                            bus.put("dropoff_distance",
ride_dropoff_object.getString("distance"));

                            buses.add(bus);
                        }

                    }
                }

                else if (ride_dropoff.length()> ride_pickup.length())
                {
                    int length_dropoff_location = ride_dropoff.length();

                    for(int j = 0; j<length_dropoff_location; j++){

                        for(int k =0; k<ride_pickup.length(); k++){
                            HashMap<String, String> bus = new HashMap<>();

                            JSONObject ride_pickup_object =
ride_pickup.getJSONObject(k);
                            JSONObject ride_dropoff_object =
ride_dropoff.getJSONObject(j);


                            bus.put("ride_date",
ride_pickup_object.getString("date"));
                            bus.put("vehicle_no_plate",
ride.getString("vehicle_no_plate"));
                            bus.put("seats_left", ride.getString("seats_left"));
                            bus.put("dropoff_location_id",
ride_dropoff_object.getString("stop_id"));

bus.put("dropoff_location",ride_dropoff_object.getString("stop_name"));
                            bus.put("dropoff_location_time",
ride_dropoff_object.getString("duration"));
                            bus.put("pickup_location_id",
ride_pickup_object.getString("stop_id"));
                            bus.put("pickup_location",
ride_pickup_object.getString("stop_name"));
                            bus.put("pickup_location_time",
ride_pickup_object.getString("duration"));
                            bus.put("arrival_time",
ride_pickup_object.getString("arrival_time"));
                            bus.put("pickup_distance",
ride_pickup_object.getString("distance"));
                            bus.put("dropoff_distance",
ride_dropoff_object.getString("distance"));

                           buses.add(bus);
                        }

                    }
                }

                else if (ride_pickup.length() == ride_dropoff.length()){

                    int length_dropoff_location = ride_dropoff.length();
                    for(int j = 0; j<length_dropoff_location; j++){
```

```
                        for(int k =0; k<ride_pickup.length(); k++){
                            HashMap<String, String> bus = new HashMap<>();

                            JSONObject ride_pickup_object =
ride_pickup.getJSONObject(k);
                            JSONObject ride_dropoff_object =
ride_dropoff.getJSONObject(j);

                            bus.put("ride_date",
ride_pickup_object.getString("date"));
                            bus.put("vehicle_no_plate",
ride.getString("vehicle_no_plate"));
                            bus.put("seats_left", ride.getString("seats_left"));
                            bus.put("dropoff_location_id",
ride_dropoff_object.getString("stop_id"));

bus.put("dropoff_location",ride_dropoff_object.getString("stop_name"));
                            bus.put("dropoff_location_time",
ride_dropoff_object.getString("duration"));
                            bus.put("pickup_location_id",
ride_pickup_object.getString("stop_id"));
                            bus.put("pickup_location",
ride_pickup_object.getString("stop_name"));
                            bus.put("pickup_location_time",
ride_pickup_object.getString("duration"));
                            bus.put("arrival_time",
ride_pickup_object.getString("arrival_time"));
                            bus.put("pickup_distance",
ride_pickup_object.getString("distance"));
                            bus.put("dropoff_distance",
ride_dropoff_object.getString("distance"));

                            buses.add(bus);
                        }

                    }
                }
            }



            list_buses.setAdapter(new CustomAdapterActivity(this, buses));
        }
        catch (Exception e){
            e.printStackTrace();
        }

        btn_back.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                onBackPressed();
            }
        });

    }

}
```

Main Activity:

```
package com.sohaibaijaz.sawaari;

import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.Navigation;

import android.content.Context;
```

```
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.StrictMode;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkResponse;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.HttpHeaderParser;
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.io.UnsupportedEncodingException;
import java.net.ConnectException;
import java.util.HashMap;
import java.util.Map;


public class MainActivity extends AppCompatActivity {

    //Shared preferences code
    public static final String AppPreferences = "AppPreferences";
    SharedPreferences sharedPreferences;
    //


    private  String token;
    private RequestQueue requestQueue;
    private EditText txt_email_phone;
    private EditText txt_password;
    private FrameLayout spinner_frame;
    private ProgressBar spinner;
    private TextView tv_forget_password;


    public static final String MAP_VIEW_BUNDLE_KEY =
"AIzaSyCxh6jiboDAWzR7c_373KDStrtj2W4Sgg4";

    public static String baseurl= "https://cc-zmac.localhost.run";
    private int backpress = 0;
    @Override
    public void onBackPressed(){
        backpress = (backpress + 1);
        Toast.makeText(getApplicationContext(), " Press Back again to Exit ",
Toast.LENGTH_SHORT).show();

        if (backpress>1) {
            this.finish();
```

```
        }
    }

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

        final Button btn_login = findViewById(R.id.btn_login);
        requestQueue = Volley.newRequestQueue(this);
        spinner = (ProgressBar)findViewById(R.id.progressBar1);
        spinner.setVisibility(View.GONE);
        txt_email_phone = findViewById(R.id.txt_email);
        txt_password = findViewById(R.id.txt_password);
        tv_forget_password = findViewById(R.id.tv_forget_password);
        spinner_frame = findViewById(R.id.spinner_frame);
        spinner_frame.setVisibility(View.GONE);



        tv_forget_password.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View view) {
                Intent i = new Intent(MainActivity.this,
ForgetPasswordActivity.class);
                MainActivity.this.startActivity(i);
            }
        });
        txt_password.setOnEditorActionListener(new
EditText.OnEditorActionListener(){

            @Override
            public boolean onEditorAction(TextView textView, int i, KeyEvent
keyEvent) {
                if (i == EditorInfo.IME_ACTION_DONE) {
                    btn_login.performClick();
                    InputMethodManager imm =
(InputMethodManager)getSystemService(Context.INPUT_METHOD_SERVICE);
                    imm.hideSoftInputFromWindow(btn_login.getWindowToken(),
                            InputMethodManager.RESULT_UNCHANGED_SHOWN);
                    return true;
                }
                return false;
            }
            });


        //Shared Preferences
        sharedPreferences = getSharedPreferences(AppPreferences,
Context.MODE_PRIVATE );


        if(!sharedPreferences.getString("Token",  "").isEmpty()){
            Intent intent = new Intent(MainActivity.this,NavActivity.class );
            finish();
            MainActivity.this.startActivity(intent);
        }

        txt_password.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (v.getId() == txt_password.getId())
                {
                    txt_password.setCursorVisible(true);
                }
            }
```

```
        });



        btn_login.setOnClickListener(btnLoginListener);


        TextView txt_signup = findViewById(R.id.txt_signup);
        txt_signup.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {

                Intent myIntent = new Intent(MainActivity.this,
SignupActivity.class);//Optional parameters
                MainActivity.this.startActivity(myIntent);
            }
        });
    }


    public View.OnClickListener btnLoginListener = new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            final String email_phone = txt_email_phone.getText().toString();
            final String password = txt_password.getText().toString();

            txt_password.setCursorVisible(false);

            if (email_phone.equals("") || password.equals("")){
                Toast.makeText(MainActivity.this, "Email or Password field empty",
Toast.LENGTH_LONG).show();
            }
            else{
                try {
                    String URL = baseurl+"/login/";
                    JSONObject jsonBody = new JSONObject();
                    jsonBody.put("email_or_phone", email_phone);
                    jsonBody.put("password", password);
                    final String requestBody = jsonBody.toString();
                    spinner.setVisibility(View.VISIBLE);
                    spinner_frame.setVisibility(View.VISIBLE);
                    StringRequest stringRequest = new
StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {

                        @Override
                        public void onResponse(String response) {
                            spinner.setVisibility(View.GONE);
                            spinner_frame.setVisibility(View.GONE);

                            Log.i("VOLLEY", response.toString());
                            try {
                                JSONObject json = new JSONObject(response);
                                if (json.getString("status").equals("200")) {
                                    token = json.getString("token");
                                    if(json.getString("message").equals("User not
authenticated. Please verify first.")){
                                        Toast.makeText(MainActivity.this,
json.getString("message"), Toast.LENGTH_SHORT).show();
                                        Intent myIntent = new
Intent(MainActivity.this, VerifyActivity.class);//Optional parameters
                                        Bundle b = new Bundle();
                                        b.putString("Token", token);
                                        b.putString("email_phone", email_phone);
                                        myIntent.putExtras(b);
                                        MainActivity.this.startActivity(myIntent);
                                    }
```

```
                                    else {
                                        //Shared Preferences
                                        SharedPreferences.Editor editor =
sharedPreferences.edit();
                                        editor.remove("Token");
                                        editor.putString("Token", token);
                                        editor.apply();

UserDetails.getUserDetails(MainActivity.this);

UserDetails.getUserRides(MainActivity.this);
                                        Intent myIntent = new
Intent(MainActivity.this, NavActivity.class);//Optional parameters
                                        finish();
                                        MainActivity.this.startActivity(myIntent);
                                    }
//


                                }
                                else if
(json.getString("status").equals("400")||json.getString("status").equals("404")) {
                                    Toast.makeText(MainActivity.this,
json.getString("message"), Toast.LENGTH_SHORT).show();
                                }
                            } catch (JSONException e) {
                                Log.e("VOLLEY", e.toString());

                            }
                        }
                    }, new Response.ErrorListener() {
                        @Override
                        public void onErrorResponse(VolleyError error) {
                            spinner.setVisibility(View.GONE);
                            spinner_frame.setVisibility(View.GONE);
                            Toast.makeText(MainActivity.this, "Server is
temporarily down, sorry for your inconvenience", Toast.LENGTH_SHORT).show();
                            Log.e("VOLLEY", error.toString());
                        }
                    }){
                        @Override
                        protected Map<String,String> getParams(){
                            Map<String,String> params = new HashMap<String,
String>();
                            params.put("email_or_phone",email_phone);
                            params.put("password",password);
//                                params.put(KEY_EMAIL, email);
                            return params;
                        }


                    };

                    stringRequest.setRetryPolicy(new RetryPolicy() {
                        @Override
                        public int getCurrentTimeout() {
                            return 50000;
                        }

                        @Override
                        public int getCurrentRetryCount() {
                            return 50000;
                        }

                        @Override
                        public void retry(VolleyError error) throws VolleyError {

                        }
```

```
                    });
                    requestQueue.add(stringRequest);


                } catch (JSONException e) {
                    e.printStackTrace();
                }

            }

        }


    };
}
```

Signup Activtiy:

```
package com.sohaibaijaz.sawaari;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.VolleyLog;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

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

import java.io.UnsupportedEncodingException;
import java.util.HashMap;
import java.util.Map;

public class SignupActivity extends AppCompatActivity {

    String phone = "";
    String email = "";
    String password = "";
    String password2 = "";
    String is_customer = "True";

    private FrameLayout spinner_frame;
    private ProgressBar spinner;
```

```
    SharedPreferences sharedPreferences;
    private int backpress = 0;
    @Override
    public void onBackPressed(){
        backpress = (backpress + 1);
        Toast.makeText(getApplicationContext(), " Press Back again to Exit ",
Toast.LENGTH_SHORT).show();

        if (backpress>1) {
            this.finish();
        }
    }

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

        sharedPreferences = getSharedPreferences(MainActivity.AppPreferences,
Context.MODE_PRIVATE );

        TextView txt_login = findViewById(R.id.txt_login);
        txt_login.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View view) {
                finish();
                return;
            }
        });

       final Button btn_signup = findViewById(R.id.btn_signup);
       final EditText txt_email = findViewById(R.id.txt_email);
       final EditText txt_phone = findViewById(R.id.txt_phone);
       final EditText txt_password = findViewById(R.id.txt_password);
       final EditText txt_password2 = findViewById(R.id.txt_password2);

        spinner = (ProgressBar)findViewById(R.id.progressBar1);
        spinner.setVisibility(View.GONE);
        spinner_frame = findViewById(R.id.spinner_frame);
        spinner_frame.setVisibility(View.GONE);
//       final EditText txt_login = findViewById(R.id.txt_login);
        txt_password2.setOnEditorActionListener(new
EditText.OnEditorActionListener(){

            @Override
            public boolean onEditorAction(TextView textView, int i, KeyEvent
keyEvent) {

                if(i== EditorInfo.IME_ACTION_DONE){
                    btn_signup.performClick();
                    InputMethodManager imm =
(InputMethodManager)getSystemService(Context.INPUT_METHOD_SERVICE);
                    imm.hideSoftInputFromWindow(btn_signup.getWindowToken(),
                            InputMethodManager.RESULT_UNCHANGED_SHOWN);
                    return true;
                }
                return false;
            }
        });
       txt_password2.setOnClickListener(new View.OnClickListener() {
           @Override
           public void onClick(View view) {
               if(view.getId() == txt_password2.getId()){
                   txt_password2.setCursorVisible(true);
               }
           }
```

```
        });

        final RequestQueue requestQueue = Volley.newRequestQueue(this);
        btn_signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                phone = txt_phone.getText().toString();
                email = txt_email.getText().toString();
                password = txt_password.getText().toString();
                password2 = txt_password2.getText().toString();


                txt_password2.setCursorVisible(false);

                if(!password.equals(password2)){
                    Toast.makeText(SignupActivity.this, "Password and Confirm
Password doesn't match", Toast.LENGTH_LONG).show();
                }


if((!phone.equals("")||!email.equals(""))&&!password.equals("")&&!password2.equals(
"")){
                    if(password.equals(password2))
                    {
                        try {
                            String URL = MainActivity.baseurl+"/register/";
                            JSONObject jsonBody = new JSONObject();
                            jsonBody.put("email", email);
                            jsonBody.put("phone_number", phone);
                            jsonBody.put("password", password);
                            jsonBody.put("confirm_password", password2);
                            jsonBody.put("is_customer", is_customer);
                            final String requestBody = jsonBody.toString();
                            spinner.setVisibility(View.VISIBLE);
                            spinner_frame.setVisibility(View.VISIBLE);
                            StringRequest stringRequest = new
StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
                                @Override
                                public void onResponse(String response) {
                                    spinner.setVisibility(View.GONE);
                                    spinner_frame.setVisibility(View.GONE);
                                    Log.i("VOLLEY", response.toString());
                                    try {

                                        JSONObject json = new JSONObject(response);
                                        if (json.getString("status").equals("200"))
{
                                            String token = json.getString("token");

                                            //Shared Preferences
                                            Toast.makeText(SignupActivity.this,
json.getString("message"), Toast.LENGTH_SHORT).show();
                                            Intent myIntent = new
Intent(SignupActivity.this, VerifyActivity.class);//Optional parameter
                                            Bundle b = new Bundle();
                                            b.putString("Token", token);
                                            myIntent.putExtras(b);

SignupActivity.this.startActivity(myIntent);

                                        }
                                        else if
(json.getString("status").equals("400")||json.getString("status").equals("404")) {
                                            Toast.makeText(SignupActivity.this,
json.getString("message"), Toast.LENGTH_SHORT).show();
                                        }
                                    } catch (JSONException e) {
```

```
                                    Log.e("VOLLEY", e.toString());

                                }
                            }
                        }, new Response.ErrorListener() {

                            @Override
                            public void onErrorResponse(VolleyError error) {
                                spinner.setVisibility(View.GONE);
                                spinner_frame.setVisibility(View.GONE);
                                Toast.makeText(SignupActivity.this, "Server is
temporarily down, sorry for your inconvenience", Toast.LENGTH_SHORT).show();
                                Log.e("VOLLEY", error.toString());
                            }
                        }){
                            @Override
                            protected Map<String,String> getParams(){
                                Map<String,String> params = new HashMap<String,
String>();

                                params.put("email",email);
                                params.put("phone_number", phone);
                                params.put("password",password);
                                params.put("confirm_password", password2);
                                params.put("is_customer", is_customer);
                                return params;
                            }
                        };

                        stringRequest.setRetryPolicy(new RetryPolicy() {
                            @Override
                            public int getCurrentTimeout() {
                                return 50000;
                            }

                            @Override
                            public int getCurrentRetryCount() {
                                return 50000;
                            }

                            @Override
                            public void retry(VolleyError error) throws
VolleyError {

                            }
                        });

                    requestQueue.add(stringRequest);

                } catch (JSONException e) {
                    e.printStackTrace();
                }
                }

            }
            else{
                Toast.makeText(SignupActivity.this, "Required fields empty!",
Toast.LENGTH_LONG).show();
            }
        }
    });


    }
}
```

Update Name Activity:

```java
package com.sohaibaijaz.sawaari;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.sohaibaijaz.sawaari.Fragments.AccountFragment;
import com.sohaibaijaz.sawaari.Fragments.HomeFragment;

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

import java.util.HashMap;
import java.util.Map;

public class UpdateNameActivity extends AppCompatActivity {



    private EditText txt_first_name;
    private EditText txt_last_name;
    private Button name_update_btn;
    SharedPreferences sharedPreferences;
    private RequestQueue requestQueue;
    private FrameLayout spinner_frame;
    private ProgressBar spinner;
    @Override
    protected void onCreate(Bundle savedInstanceState) {


        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_name);
        getSupportActionBar().hide();
        sharedPreferences = getSharedPreferences(MainActivity.AppPreferences,
Context.MODE_PRIVATE );
        final String token = sharedPreferences.getString("Token", "");
        spinner = (ProgressBar)findViewById(R.id.progressBar1);
        spinner.setVisibility(View.GONE);
        spinner_frame = findViewById(R.id.spinner_frame);
        spinner_frame.setVisibility(View.GONE);

        txt_first_name =(EditText) findViewById(R.id.txt_first_name);
        txt_last_name = (EditText)findViewById(R.id.txt_last_name);
        name_update_btn = (Button)findViewById(R.id.name_update_btn);
```

```
        txt_last_name.setOnEditorActionListener(new
EditText.OnEditorActionListener(){

            @Override
            public boolean onEditorAction(TextView textView, int i, KeyEvent
keyEvent) {
                if(i== EditorInfo.IME_ACTION_DONE || i== KeyEvent.KEYCODE_ENTER){
                    name_update_btn.performClick();
                    InputMethodManager imm =
(InputMethodManager)getSystemService(Context.INPUT_METHOD_SERVICE);
                    imm.hideSoftInputFromWindow(name_update_btn.getWindowToken(),
                            InputMethodManager.RESULT_UNCHANGED_SHOWN);
                    return true;
                }

                return false;
            }
        });



        name_update_btn.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View view) {

                final String first_name = txt_first_name.getText().toString();
                final String last_name = txt_last_name.getText().toString();


                if (first_name.equals("") || last_name.equals("")) {
                    Toast.makeText(getApplicationContext(), "You must enter First
name and Last name!", Toast.LENGTH_SHORT).show();
                }
                else{
                requestQueue = Volley.newRequestQueue(getApplicationContext());
                try {
                    String URL = MainActivity.baseurl + "/update/name/";
                    spinner.setVisibility(View.VISIBLE);
                    spinner_frame.setVisibility(View.VISIBLE);
                    StringRequest stringRequest = new
StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
                        @Override
                        public void onResponse(String response) {
                            spinner.setVisibility(View.GONE);
                            spinner_frame.setVisibility(View.GONE);
                            Log.i("VOLLEY", response.toString());
                            try {
                                JSONObject json = new JSONObject(response);
                                if (json.getString("status").equals("200")) {
                                    Toast.makeText(getApplicationContext(),
json.getString("message"), Toast.LENGTH_SHORT).show();

UserDetails.getUserDetails(UpdateNameActivity.this);
                                    Intent i = new Intent(getApplicationContext(),
MainActivity.class);
                                    startActivity(i);
                                    finish();
                                } else if (json.getString("status").equals("400")
|| json.getString("status").equals("404")) {
                                    Toast.makeText(getApplicationContext(),
json.getString("message"), Toast.LENGTH_SHORT).show();
                                }
                            } catch (JSONException e) {
                                Log.e("VOLLEY", e.toString());

                            }
```

```
                        }
                    }, new Response.ErrorListener() {
                        @Override
                        public void onErrorResponse(VolleyError error) {
                            spinner.setVisibility(View.GONE);
                            spinner_frame.setVisibility(View.GONE);
                            Toast.makeText(getApplicationContext(), "Server is
temporarily down, sorry for your inconvenience", Toast.LENGTH_SHORT).show();
                            Log.e("VOLLEY", error.toString());
                        }
                    }) {
                        @Override
                        protected Map<String, String> getParams() {
                            Map<String, String> params = new HashMap<String,
String>();
                            params.put("first_name", first_name);
                            params.put("last_name", last_name);
                            return params;
                        }

                        @Override
                        public Map<String, String> getHeaders() throws
AuthFailureError {
                            Map<String, String> params = new HashMap<String,
String>();
                            params.put("Authorization", token);
                            return params;
                        }


                    };

                    stringRequest.setRetryPolicy(new RetryPolicy() {
                        @Override
                        public int getCurrentTimeout() {
                            return 50000;
                        }

                        @Override
                        public int getCurrentRetryCount() {
                            return 50000;
                        }

                        @Override
                        public void retry(VolleyError error) throws VolleyError {

                        }
                    });
                    requestQueue.add(stringRequest);

                } catch (Exception e) {
                    e.printStackTrace();
                }


            }
            }
        });
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent i = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(i);
    }
}
```

Update Phone Number Activity:

```
package com.sohaibaijaz.sawaari;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.KeyEvent;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

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

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

import java.util.HashMap;
import java.util.Map;

public class UpdatePhoneNumberActivity extends AppCompatActivity {


    private EditText txt_phone_number;
    private EditText txt_confirm_phone_number;
    private Button phone_update_btn;
    SharedPreferences sharedPreferences;
    private RequestQueue requestQueue;
    private FrameLayout spinner_frame;
    private ProgressBar spinner;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_update_phone_number);
        getSupportActionBar().hide();

        sharedPreferences = getSharedPreferences(MainActivity.AppPreferences,
Context.MODE_PRIVATE );
        final String token = sharedPreferences.getString("Token", "");
        spinner = (ProgressBar)findViewById(R.id.progressBar1);
        spinner.setVisibility(View.GONE);
        spinner_frame = findViewById(R.id.spinner_frame);
        spinner_frame.setVisibility(View.GONE);

        txt_phone_number =(EditText) findViewById(R.id.txt_phone_number);
        txt_confirm_phone_number =
(EditText)findViewById(R.id.txt_confirm_phone_number);
        phone_update_btn = (Button)findViewById(R.id.phone_update_btn);
```

```
        txt_confirm_phone_number.setOnEditorActionListener(new
EditText.OnEditorActionListener(){

            @Override
            public boolean onEditorAction(TextView textView, int i, KeyEvent
keyEvent) {
                if(i== EditorInfo.IME_ACTION_DONE || i== KeyEvent.KEYCODE_ENTER){
                    phone_update_btn.performClick();
                    InputMethodManager imm =
(InputMethodManager)getSystemService(Context.INPUT_METHOD_SERVICE);
                    imm.hideSoftInputFromWindow(phone_update_btn.getWindowToken(),
                            InputMethodManager.RESULT_UNCHANGED_SHOWN);
                    return true;
                }

                return false;
            }
        });


        phone_update_btn.setOnClickListener(new View.OnClickListener(){

            @Override
            public void onClick(View view) {

                final String phone_number = txt_phone_number.getText().toString();
                final String confirm_phone_number =
txt_confirm_phone_number.getText().toString();

               if(!phone_number.equals(confirm_phone_number)){
                   Toast.makeText(getApplicationContext(), "Both fields doesn't
match",Toast.LENGTH_SHORT).show();
                }
                else {

                    requestQueue = Volley.newRequestQueue(getApplicationContext());
                    try {
                        String URL = MainActivity.baseurl + "/change/phonenumber/";
                        spinner.setVisibility(View.VISIBLE);
                        spinner_frame.setVisibility(View.VISIBLE);
                        StringRequest stringRequest = new
StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
                            @Override
                            public void onResponse(String response) {
                                spinner.setVisibility(View.GONE);
                                spinner_frame.setVisibility(View.GONE);
                                Log.i("VOLLEY", response.toString());
                                try {
                                    JSONObject json = new JSONObject(response);
                                    if (json.getString("status").equals("200")) {
                                        Toast.makeText(getApplicationContext(),
json.getString("message"), Toast.LENGTH_SHORT).show();
                                        finish();
                                        Intent i = new
Intent(getApplicationContext(), VerifyPhoneNumberActivity.class);
                                        i.putExtra("phone_number", phone_number);
                                        startActivity(i);
                                    } else if
(json.getString("status").equals("400") || json.getString("status").equals("404"))
{
                                        Toast.makeText(getApplicationContext(),
json.getString("message"), Toast.LENGTH_SHORT).show();
                                    }
                                } catch (JSONException e) {
                                    Log.e("VOLLEY", e.toString());
```

```
                                }
                            }
                        }, new Response.ErrorListener() {
                            @Override
                            public void onErrorResponse(VolleyError error) {
                                spinner.setVisibility(View.GONE);
                                spinner_frame.setVisibility(View.GONE);
                                Toast.makeText(getApplicationContext(), "Server is
temporarily down, sorry for your inconvenience", Toast.LENGTH_SHORT).show();
                                Log.e("VOLLEY", error.toString());
                            }
                        }) {
                            @Override
                            protected Map<String, String> getParams() {
                                Map<String, String> params = new HashMap<String,
String>();
                                params.put("phonenumber", phone_number);
                                return params;
                            }

                            @Override
                            public Map<String, String> getHeaders() throws
AuthFailureError {
                                Map<String, String> params = new HashMap<String,
String>();
                                params.put("Authorization", token);
                                return params;
                            }


                        };

                        stringRequest.setRetryPolicy(new RetryPolicy() {
                            @Override
                            public int getCurrentTimeout() {
                                return 50000;
                            }

                            @Override
                            public int getCurrentRetryCount() {
                                return 50000;
                            }

                            @Override
                            public void retry(VolleyError error) throws VolleyError
{

                            }
                        });
                        requestQueue.add(stringRequest);

                    } catch (Exception e) {
                        e.printStackTrace();
                    }

                }
            }
        });


    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent i = new Intent(getApplicationContext(), MainActivity.class);
        startActivity(i);
```

```
    }
}
```

User Details:

```
package com.sohaibaijaz.sawaari;

import android.content.Context;
import android.content.SharedPreferences;
import android.util.Log;
import android.widget.Toast;

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

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

import java.util.HashMap;
import java.util.Map;
import java.util.Objects;

public class UserDetails {


    public static void getUserDetails(final Context context){
        final SharedPreferences sharedPreferences=
Objects.requireNonNull(context).getSharedPreferences(MainActivity.AppPreferences,
Context.MODE_PRIVATE);
        final RequestQueue requestQueue = Volley.newRequestQueue(context);
        final String token = sharedPreferences.getString("Token", "");

        if(!token.equals("")) {

           //Getting user details
            try {
                String URL = MainActivity.baseurl + "/my_details/";
                JSONObject jsonBody = new JSONObject();
                StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL, new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        Log.i("VOLLEY", response.toString());
                        try {
                            JSONObject json = new JSONObject(response);

                            if (json.getString("status").equals("200")) {
                                SharedPreferences.Editor edit =
sharedPreferences.edit();
                                edit.putString("first_name",
json.getString("first_name"));
                                edit.putString("last_name",
json.getString("last_name"));
                                edit.putString("email", json.getString("email"));
                                edit.putString("phone_number",
json.getString("phone_number"));
                                edit.apply();
                            } else if (json.getString("status").equals("400") ||
json.getString("status").equals("404") || json.getString("status").equals("405")) {
```

```
                                Toast.makeText(context, json.getString("message"),
Toast.LENGTH_SHORT).show();
                            }

                        } catch (JSONException e) {
                            Log.e("VOLLEY", e.toString());
                        }
                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(context, "Server is temporarily down, sorry
for your inconvenience", Toast.LENGTH_SHORT).show();
                        Log.e("VOLLEY", error.toString());
                    }
                }) {
                    @Override
                    protected Map<String, String> getParams() {
                        Map<String, String> params = new HashMap<String, String>();


                        return params;
                    }

                    @Override
                    public Map<String, String> getHeaders() throws AuthFailureError
{
                        Map<String, String> headers = new HashMap<String,
String>();
                        headers.put("Authorization", token);
                        return headers;
                    }
                };

                stringRequest.setRetryPolicy(new RetryPolicy() {
                    @Override
                    public int getCurrentTimeout() {
                        return 50000;
                    }

                    @Override
                    public int getCurrentRetryCount() {
                        return 50000;
                    }

                    @Override
                    public void retry(VolleyError error) throws VolleyError {

                    }
                });

                requestQueue.add(stringRequest);
            } catch (Exception e) {
                Toast.makeText(context, "Slow Internet Connection.",
Toast.LENGTH_SHORT).show();
            }




        }
        else{
            Toast.makeText(context, "There was problem connecting to the server",
Toast.LENGTH_SHORT).show();
        }
    }
```

```
    public static void getUserRides(final Context context){
        final SharedPreferences sharedPreferences=
Objects.requireNonNull(context).getSharedPreferences(MainActivity.AppPreferences,
Context.MODE_PRIVATE);
        final RequestQueue requestQueue = Volley.newRequestQueue(context);
        final String token = sharedPreferences.getString("Token", "");

        if(!token.equals("")) {

            //Getting user rides
            try {
                String URL = MainActivity.baseurl + "/user_rides/";
                JSONObject jsonBody = new JSONObject();
                StringRequest stringRequest = new StringRequest(Request.Method.GET,
URL, new Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {

                        Log.i("VOLLEY", response.toString());
                        try {
                            JSONObject json = new JSONObject(response);

                            if (json.getString("status").equals("200")) {
                                SharedPreferences.Editor edit =
sharedPreferences.edit();
                                edit.putString("user_rides",
json.getJSONArray("reservations").toString());
                                edit.apply();
                            } else if (json.getString("status").equals("400") ||
json.getString("status").equals("404") || json.getString("status").equals("405")) {
                                Toast.makeText(context, json.getString("message"),
Toast.LENGTH_SHORT).show();
                            }

                        } catch (JSONException e) {
                            Log.e("VOLLEY", e.toString());
                        }
                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {
                        Toast.makeText(context, "Server is temporarily down, sorry
for your inconvenience", Toast.LENGTH_SHORT).show();
                        Log.e("VOLLEY", error.toString());
                    }
                }) {
                    @Override
                    protected Map<String, String> getParams() {
                        Map<String, String> params = new HashMap<String, String>();


                        return params;
                    }

                    @Override
                    public Map<String, String> getHeaders() throws AuthFailureError
{
                        Map<String, String> headers = new HashMap<String,
String>();
                        headers.put("Authorization", token);
                        return headers;
                    }
                };

                stringRequest.setRetryPolicy(new RetryPolicy() {
                    @Override
                    public int getCurrentTimeout() {
                        return 50000;
```

```
                    }

                    @Override
                    public int getCurrentRetryCount() {
                        return 50000;
                    }

                    @Override
                    public void retry(VolleyError error) throws VolleyError {

                    }
                });

                requestQueue.add(stringRequest);
            } catch (Exception e) {
                Toast.makeText(context, "Slow Internet Connection.",
Toast.LENGTH_SHORT).show();
            }


        }
        else{
            Toast.makeText(context, "There was problem connecting to the server",
Toast.LENGTH_SHORT).show();
        }
    }



}
```

API's built on Python (Django Rest Framework):

Admin Login API 'https://127.0.0.1:8000/admin/'

Login API 'https://127.0.0.1:8000/login/'
Logout API 'https://127.0.0.1:8000/logout/'

Registration API 'https://127.0.0.1:8000/register'
Verify User API 'https://127.0.0.1:8000/is_verified'
Register Resend Otp API 'https://127.0.0.1:8000/register/resend_otp'

Update Name API 'https://127.0.0.1:8000/update/name'
Password Change API 'https://127.0.0.1:8000/password_change/'
Phone Number Change API 'https://127.0.0.1:8000/change/phonenumber/'
Verify Phone Number API 'https://127.0.0.1:8000/verify/phonenumber/'

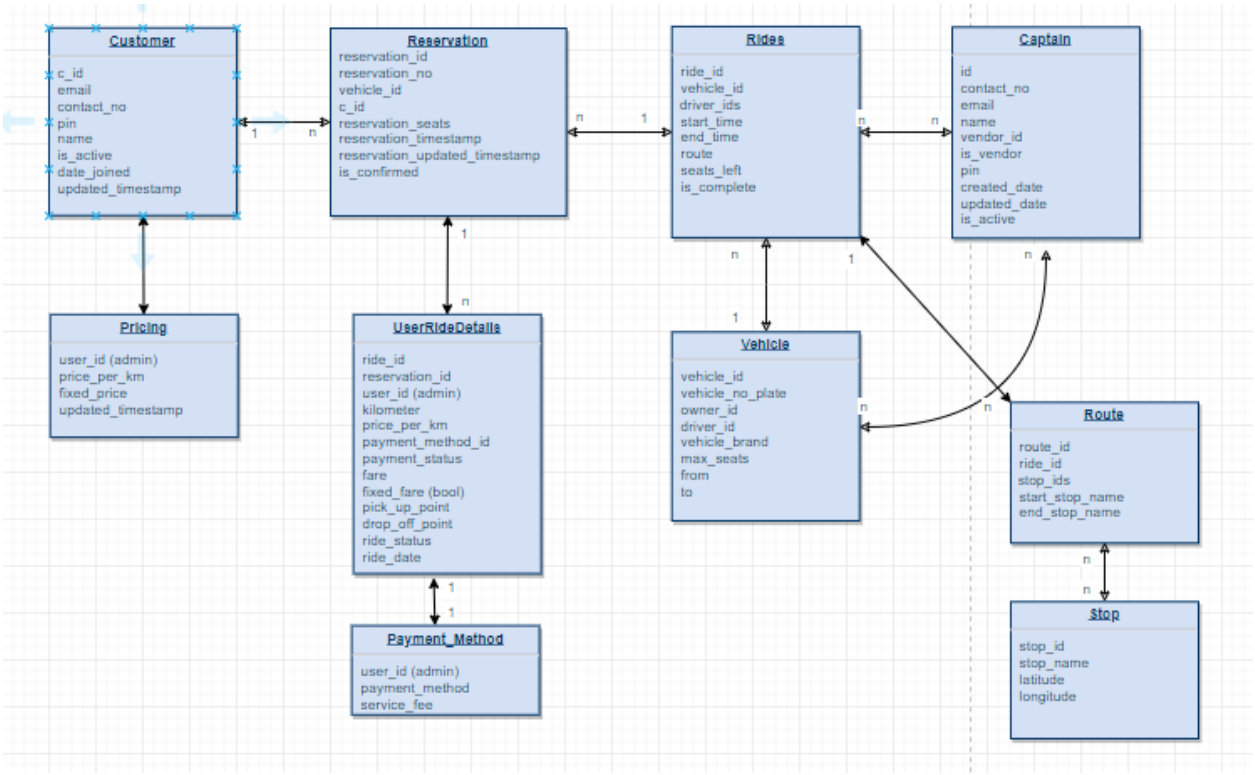
Password Reset API 'https://127.0.0.1:8000/password/reset/'
Password Reset Resend Otp API 'https://127.0.0.1:8000/password/reset/resend_otp'
Confirm Password Reset API 'https://127.0.0.1:8000/confirm/password/reset/'
New Password Reset API 'https://127.0.0.1:8000/new/password/reset/'

User Rides API 'https://127.0.0.1:8000/user_rides/'
User Details API 'https://127.0.0.1:8000/my_details/'

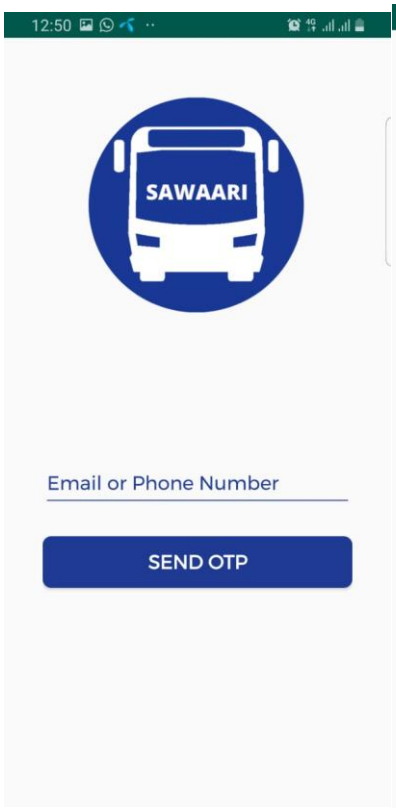
Bus Route API 'https://127.0.0.1:8000/bus/route/'
Book Ride API 'https://127.0.0.1:8000/book_ride/'

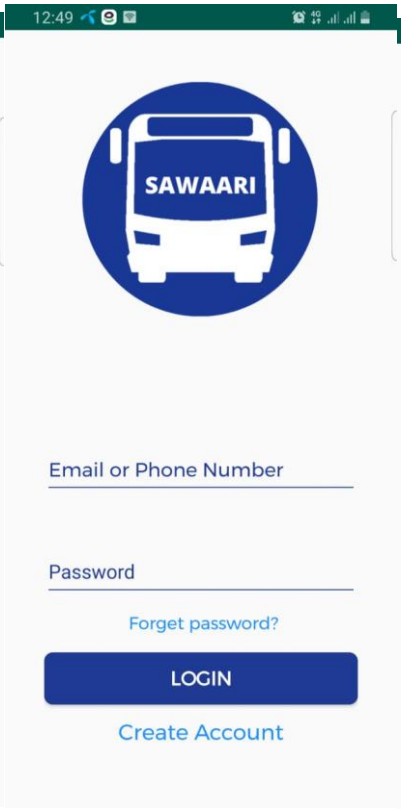
Confirm Ride API ‘https://127.0.0.1:8000/confirm_ride/’
Cancel Ride API ‘https://127.0.0.1:8000/cancel_ride/’

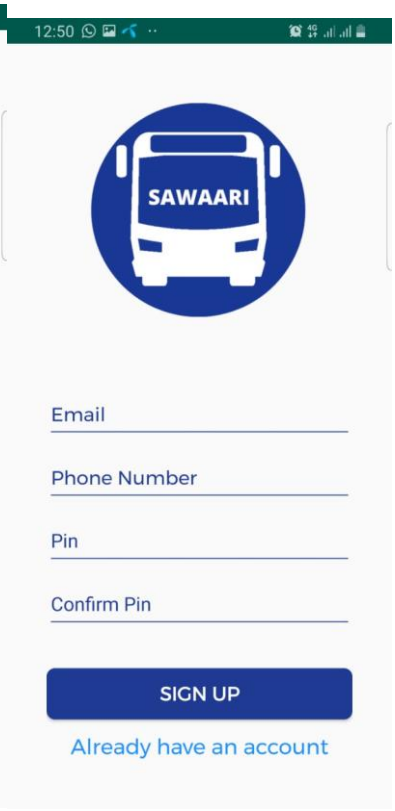
ERD:

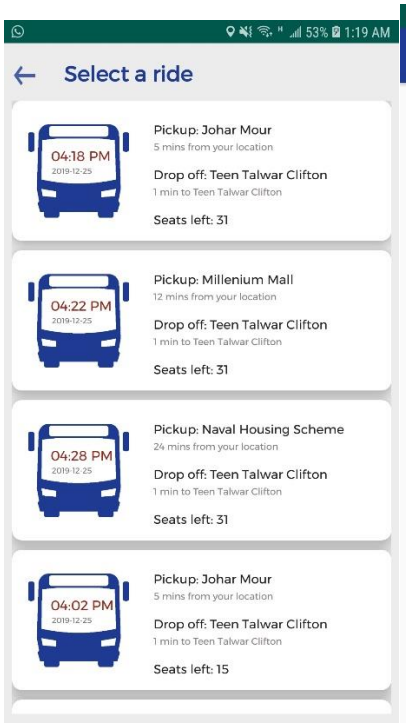


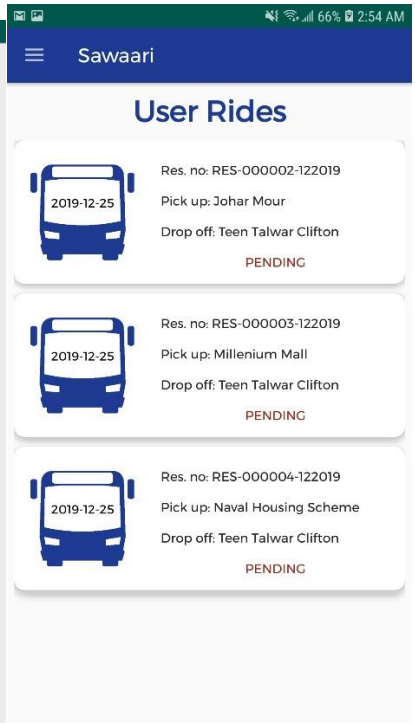
OUTPUT:

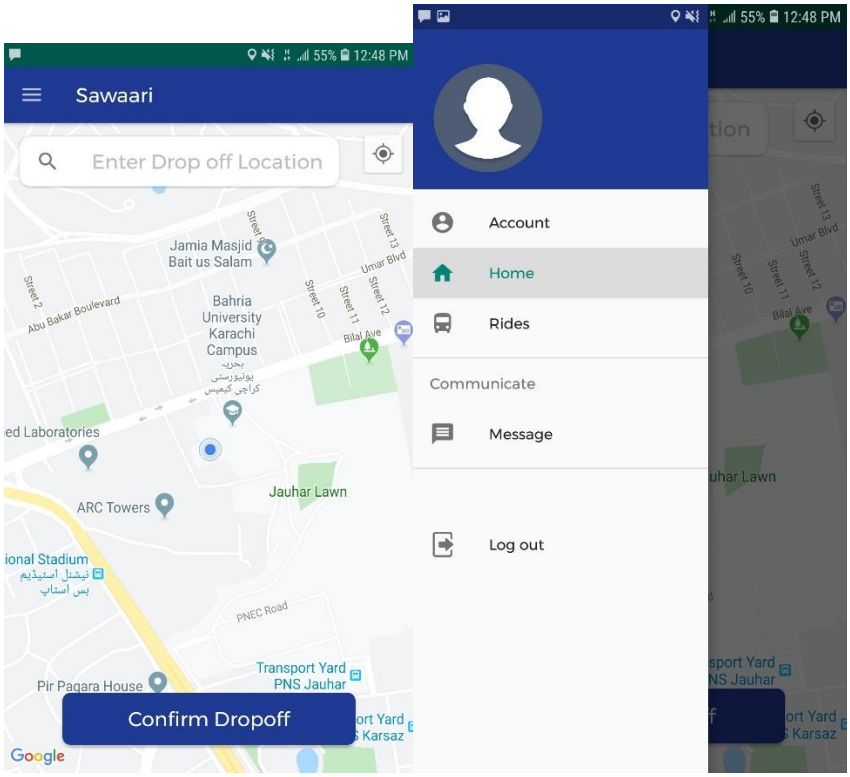

12:50
SAWAARI
Email or Phone Number
SEND OTP

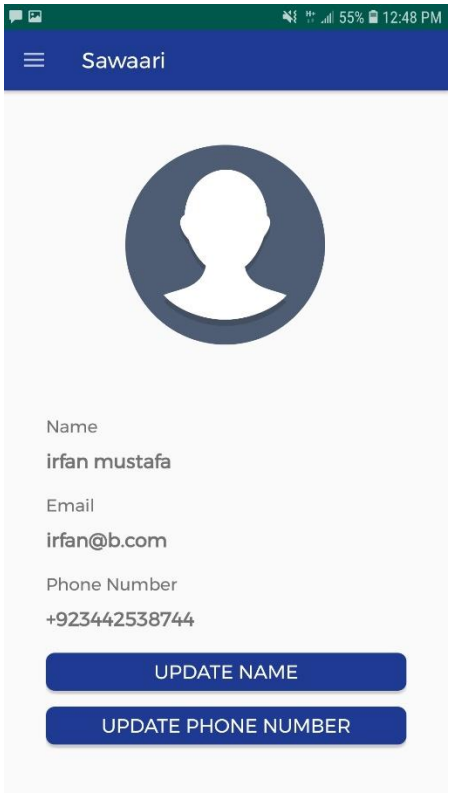

12:49
SAWAARI
Email or Phone Number
Password
Forget password?
LOGIN
Create Account


12:50
SAWAARI
Email
Phone Number
Pin
Confirm Pin
SIGN UP
Already have an account


53% 1:19 AM
Select a ride
04:18 PM
2019-12-25
Pickup: Johar Mour
5 mins from your location
Drop off: Teen Talwar Clifton
1 min to Teen Talwar Clifton
Seats left: 31
04:22 PM
2019-12-25
Pickup: Millenium Mall
12 mins from your location
Drop off: Teen Talwar Clifton
1 min to Teen Talwar Clifton
Seats left: 31
04:28 PM
2019-12-25
Pickup: Naval Housing Scheme
24 mins from your location
Drop off: Teen Talwar Clifton
1 min to Teen Talwar Clifton
Seats left: 31
04:02 PM
2019-12-25
Pickup: Johar Mour
5 mins from your location
Drop off: Teen Talwar Clifton
1 min to Teen Talwar Clifton
Seats left: 15


66% 2:54 AM
Sawaari
User Rides
2019-12-25
Res. no: RES-000002-122019
Pick up: Johar Mour
Drop off: Teen Talwar Clifton
PENDING
2019-12-25
Res. no: RES-000003-122019
Pick up: Millenium Mall
Drop off: Teen Talwar Clifton
PENDING
2019-12-25
Res. no: RES-000004-122019
Pick up: Naval Housing Scheme
Drop off: Teen Talwar Clifton
PENDING





REFERENCES:

- Airlift.
- Swvl.
- Careem.