**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:**

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/](http://127.0.0.1:8000/rest-auth/password/reset/)update/name’

Password Change API ‘<https://127.0.0.1:8000/pa>ssword\_change/’

Phone Number Change API ‘https://127.0.0.1:8000/change/phonenumber/’

Verify Phone Number API ‘[https://127.0.0.1:8000/](https://127.0.0.1:8000/pa)verify/phonenumber/’

Password Reset API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)password/reset/’

Password Reset Resend Otp API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)password/reset/resend\_otp’

Confirm Password Reset API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)confirm/password/reset/’

New Password Reset API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)new/password/reset/’

User Rides API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)user\_rides/’

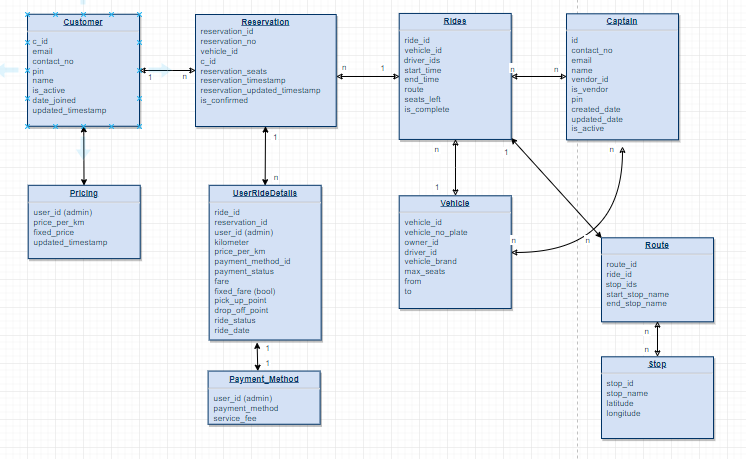
User Details API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)my\_details/’

Bus Route API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)bus/route/’

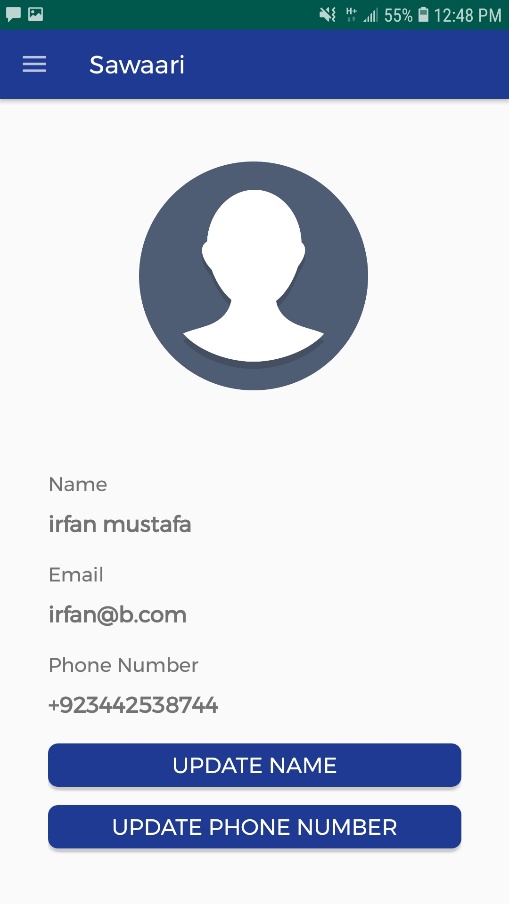
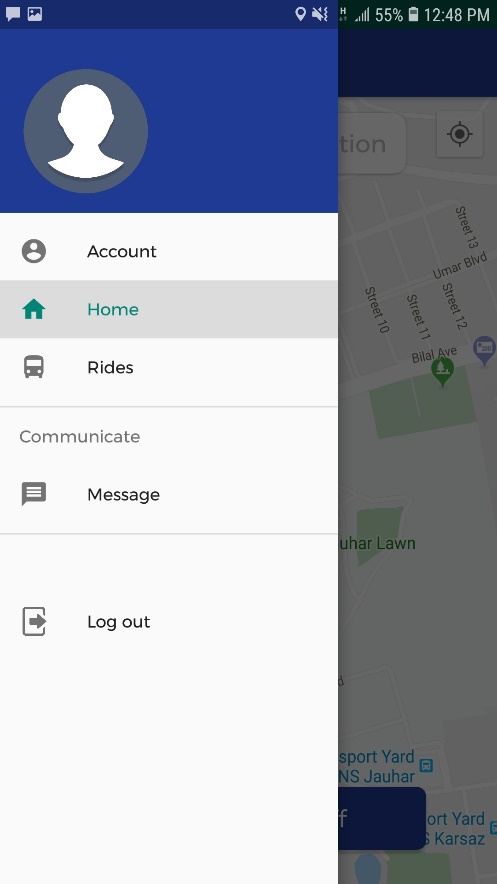
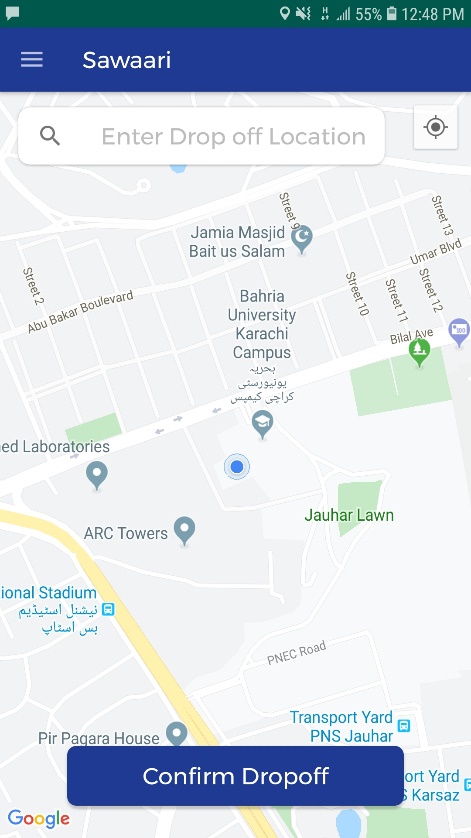
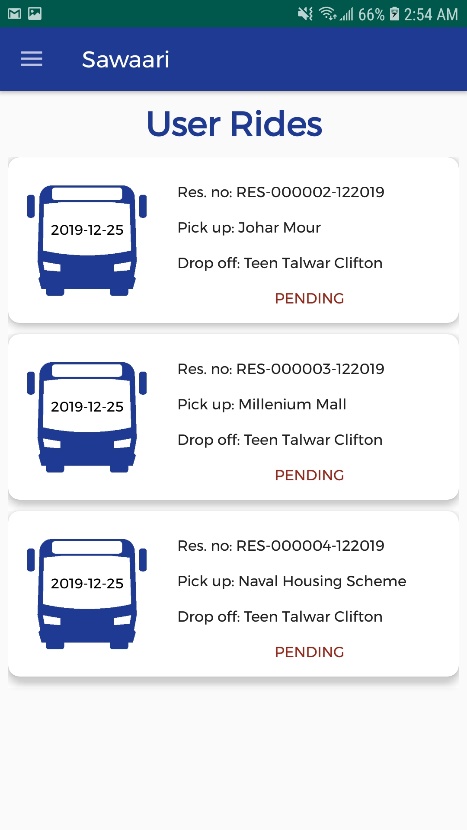
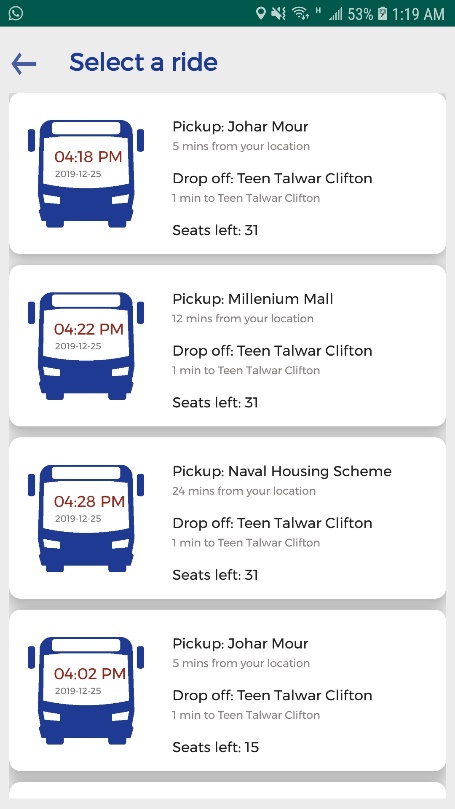
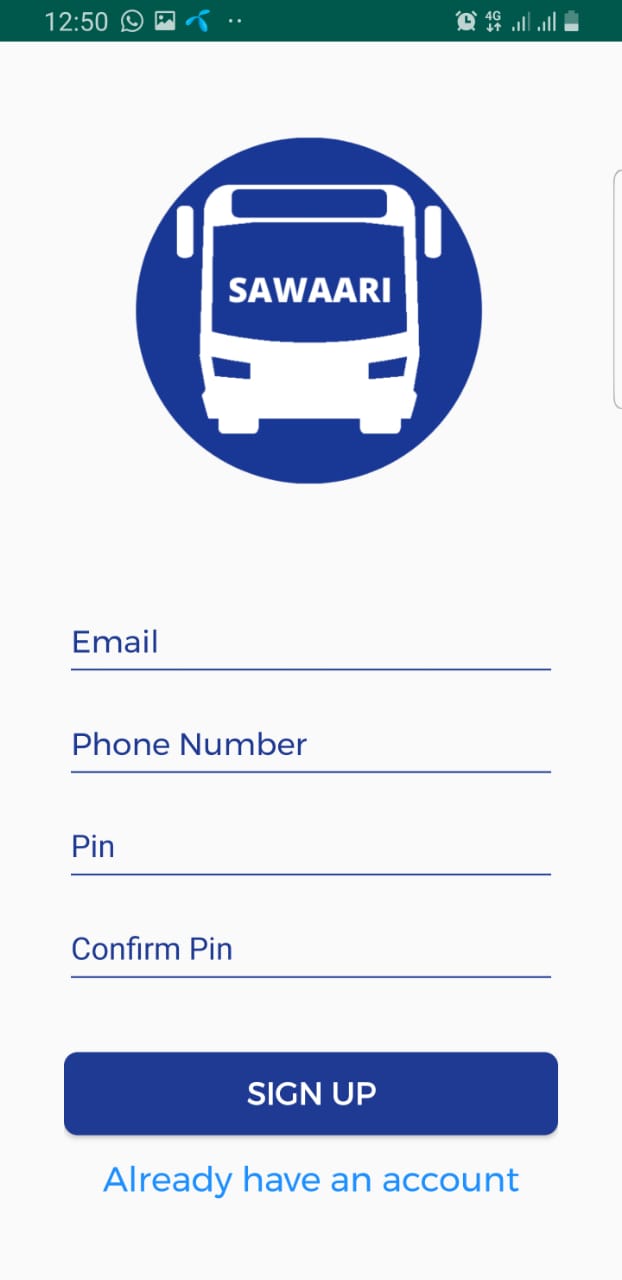
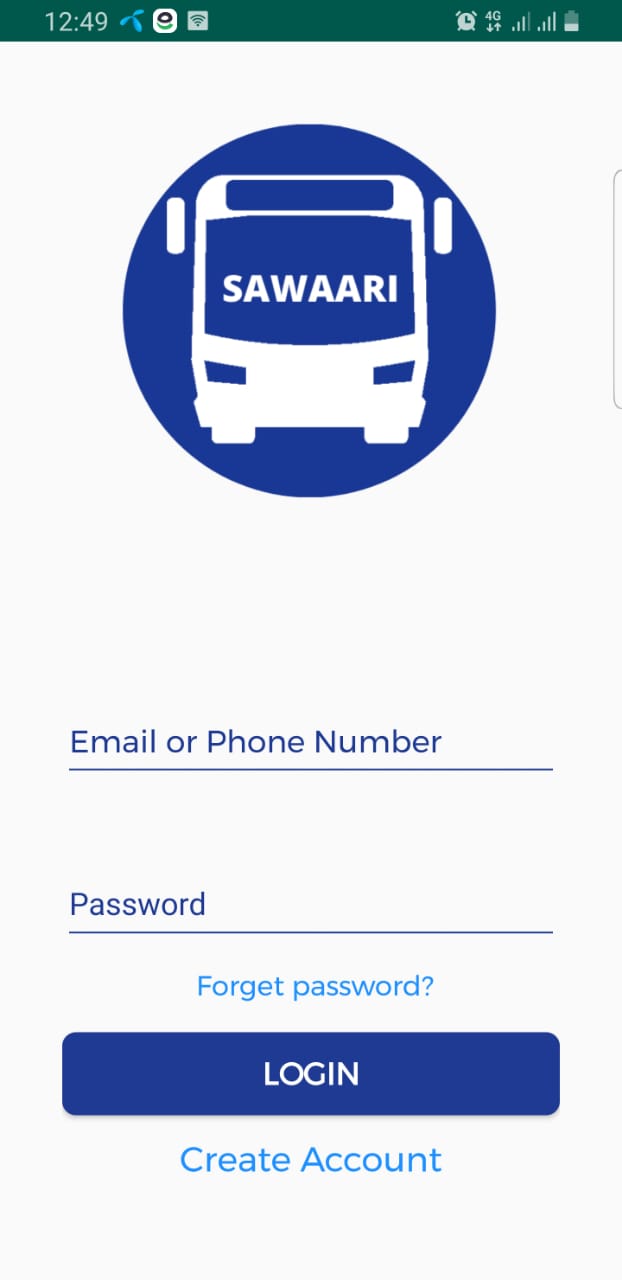
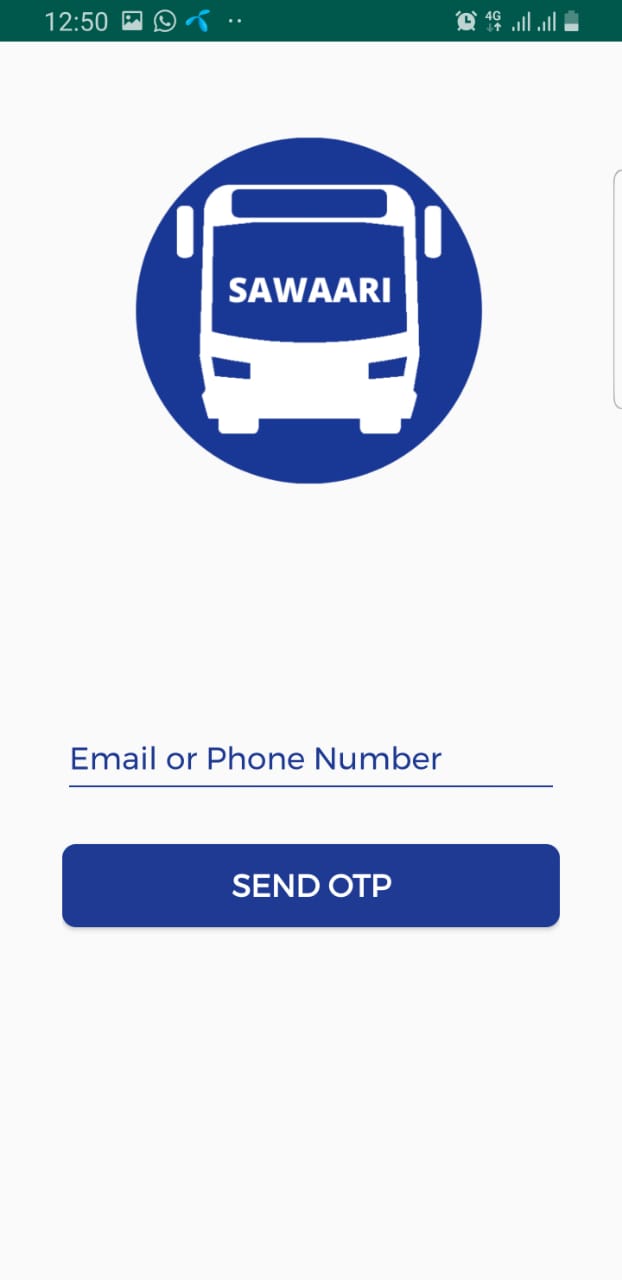
Book Ride API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)book\_ride/’

Confirm Ride API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)confirm\_ride/’

Cancel Ride API ‘[https://127.0.0.1:8000/](http://127.0.0.1:8000/rest-auth/password/reset/)cancel\_ride/’

**ERD:**

**OUTPUT:**

****

**REFERENCES:**

* Airlift.
* Swvl.
* Careem.