Sign\_up.java

**package in.skylinelabs.kym;**

**import android.Manifest;**

**import android.app.AlertDialog;**

**import android.content.DialogInterface;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.content.pm.PackageManager;**

**import android.os.Build;**

**import android.os.Bundle;**

**import android.provider.Settings;**

**import android.support.annotation.NonNull;**

**import android.support.design.widget.Snackbar;**

**import android.support.design.widget.TextInputLayout;**

**import android.support.v4.app.ActivityCompat;**

**import android.support.v4.content.ContextCompat;**

**import android.support.v7.app.AppCompatActivity;**

**import android.view.View;**

**import android.widget.Button;**

**import android.widget.EditText;**

**import com.google.gson.JsonObject;**

**import org.json.JSONException;**

**import java.util.ArrayList;**

**import java.util.List;**

**/\*\***

**\* Created by Jay Lohokare on 16-Oct-16.**

**\*/**

**public class Sign\_up extends AppCompatActivity implements AsyncTaskComplete {**

**Button btn;**

**EditText edttxt1,edttxt2, edttxt3, edttxt4;**

**TextInputLayout lNameLayout,lNameLayout1,lNameLayout2,lNameLayout4;**

**String name;**

**AlertDialog alertDialog;**

**private ActionHandler actionHandler;**

**final String PREFS\_NAME = "Kym\_App\_Preferences";**

**SharedPreferences preferences;**

**private static final String [] DANGEROUS\_PERMISSIONS = {**

**Manifest.permission.READ\_PHONE\_STATE,**

**Manifest.permission.READ\_SMS,**

**Manifest.permission.READ\_EXTERNAL\_STORAGE**

**};**

**private void initPermissions() {**

**List<String> missingPermissions = new ArrayList<String>();**

**for(String permission : DANGEROUS\_PERMISSIONS) {**

**if(ActivityCompat.checkSelfPermission(this, permission) != PackageManager.PERMISSION\_GRANTED) {**

**missingPermissions.add(permission);**

**}**

**}**

**if (missingPermissions.size() > 0) {**

**String [] permissions = new String[missingPermissions.size()];**

**ActivityCompat.requestPermissions(**

**this,**

**missingPermissions.toArray(permissions),**

**1);**

**} else {**

**// we have all permissions, move on**

**}**

**}**

**@Override**

**public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {**

**super.onRequestPermissionsResult(requestCode, permissions, grantResults);**

**for(int grantResult : grantResults) {**

**// handle denied permissions**

**}**

**}**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_sign\_up);**

**initPermissions();**

**preferences = getSharedPreferences(PREFS\_NAME, 0);**

**actionHandler = new ActionHandler(this, Sign\_up.this);**

**lNameLayout = (TextInputLayout) findViewById(R.id**

**.fNameLayoutName);**

**lNameLayout1 = (TextInputLayout) findViewById(R.id**

**.fNameLayoutContact);**

**lNameLayout4 = (TextInputLayout) findViewById(R.id**

**.fNameLayoutEmail);**

**btn = (Button) findViewById(R.id.textView5);**

**edttxt1 = (EditText) findViewById(R.id.editTextName);//Name**

**edttxt2 = (EditText) findViewById(R.id.editTextTwitter);//twitter**

**edttxt4 = (EditText) findViewById(R.id.editTextEmail);//Email**

**edttxt2.clearFocus();**

**edttxt4.clearFocus();**

**edttxt1.requestFocus();**

**btn.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**final String PREFS\_NAME = "Kym\_App\_Preferences";**

**final SharedPreferences settings = getSharedPreferences(PREFS\_NAME, 0);**

**name = edttxt1.getText().toString();**

**String email = edttxt4.getText().toString();**

**String twitter = edttxt2.getText().toString();**

**lNameLayout.setErrorEnabled(false);**

**lNameLayout1.setErrorEnabled(false);**

**lNameLayout4.setErrorEnabled(false);**

**if (edttxt1.getText().toString().matches("") || edttxt4.getText().toString().matches("") || edttxt2.getText().toString().matches("")) {**

**if (edttxt1.getText().toString().matches("")) {**

**lNameLayout.setErrorEnabled(true);**

**lNameLayout.setError("\*Required field");**

**}**

**if (edttxt4.getText().toString().matches("")) {**

**lNameLayout4.setErrorEnabled(true);**

**lNameLayout4.setError("\*Required field");**

**}**

**if (edttxt2.getText().toString().matches("")) {**

**lNameLayout1.setErrorEnabled(true);**

**lNameLayout1.setError("\*Required field");**

**}**

**} else {**

**settings.edit().putString("name", name).commit();**

**settings.edit().putString("email", email).commit();**

**settings.edit().putString("twitter\_handle", twitter).commit();**

**actionHandler.login(name, twitter);**

**}**

**}**

**});**

**}**

**@Override**

**public void handleResult(JsonObject result, String action) throws JSONException {**

**System.out.println("This is the result" + result);**

**if (result.get("success").getAsInt() == -1) {**

**Snackbar snackbar = Snackbar.make(findViewById(R.id.rel2), "No connection", Snackbar.LENGTH\_LONG);**

**snackbar.show();**

**return;**

**}**

**switch (action) {**

**case "Login":**

**if (result.get("success").getAsInt() == 1) {**

**preferences.edit().putBoolean("logged\_in", true).apply();**

**Intent intent = new Intent(this, MainActivity.class);**

**startActivity(intent);**

**finish();**

**}**

**else if(result.get("success").getAsInt() == 2){**

**AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);**

**alertDialogBuilder.setMessage("Unable to signup");**

**alertDialogBuilder.setTitle("Milala re");**

**alertDialogBuilder.setCancelable(true);**

**alertDialogBuilder.setNegativeButton("OK",**

**new DialogInterface.OnClickListener() {**

**@Override**

**public void onClick(DialogInterface dialog, int which) {**

**}**

**});**

**alertDialog = alertDialogBuilder.create();**

**alertDialog.show();**

**}**

**else {**

**AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);**

**alertDialogBuilder.setMessage("Unable to signup");**

**alertDialogBuilder.setTitle("Error");**

**alertDialogBuilder.setCancelable(true);**

**alertDialogBuilder.setNegativeButton("OK",**

**new DialogInterface.OnClickListener() {**

**@Override**

**public void onClick(DialogInterface dialog, int which) {**

**}**

**});**

**alertDialog = alertDialogBuilder.create();**

**alertDialog.show();**

**}**

**break;**

**}**

**}**

**}**

**MainActivity.java**

**package in.skylinelabs.kym;**

**import android.content.Intent;**

**import android.os.Bundle;**

**import android.support.design.widget.FloatingActionButton;**

**import android.support.design.widget.Snackbar;**

**import android.view.View;**

**import android.support.design.widget.NavigationView;**

**import android.support.v4.view.GravityCompat;**

**import android.support.v4.widget.DrawerLayout;**

**import android.support.v7.app.ActionBarDrawerToggle;**

**import android.support.v7.app.AppCompatActivity;**

**import android.support.v7.widget.Toolbar;**

**import android.view.Menu;**

**import android.view.MenuItem;**

**public class MainActivity extends AppCompatActivity**

**implements NavigationView.OnNavigationItemSelectedListener {**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);**

**setSupportActionBar(toolbar);**

**FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);**

**fab.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH\_LONG)**

**.setAction("Action", null).show();**

**}**

**});**

**DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);**

**ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(**

**this, drawer, toolbar, R.string.navigation\_drawer\_open, R.string.navigation\_drawer\_close);**

**drawer.setDrawerListener(toggle);**

**toggle.syncState();**

**NavigationView navigationView = (NavigationView) findViewById(R.id.nav\_view);**

**navigationView.setNavigationItemSelectedListener(this);**

**}**

**@Override**

**public void onBackPressed() {**

**DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);**

**if (drawer.isDrawerOpen(GravityCompat.START)) {**

**drawer.closeDrawer(GravityCompat.START);**

**} else {**

**super.onBackPressed();**

**}**

**}**

**@Override**

**public boolean onCreateOptionsMenu(Menu menu) {**

**// Inflate the menu; this adds items to the action bar if it is present.**

**getMenuInflater().inflate(R.menu.main, menu);**

**return true;**

**}**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**// Handle action bar item clicks here. The action bar will**

**// automatically handle clicks on the Home/Up button, so long**

**// as you specify a parent activity in AndroidManifest.xml.**

**int id = item.getItemId();**

**//noinspection SimplifiableIfStatement**

**if (id == R.id.action\_settings) {**

**return true;**

**}**

**return super.onOptionsItemSelected(item);**

**}**

**@SuppressWarnings("StatementWithEmptyBody")**

**@Override**

**public boolean onNavigationItemSelected(MenuItem item) {**

**// Handle navigation view item clicks here.**

**int id = item.getItemId();**

**if (id == R.id.home) {**

**// Handle the camera action**

**} else if (id == R.id.chat\_bot) {**

**Intent i = new Intent(this, ChatActivity.class);**

**startActivity(i);**

**} else if (id == R.id.tutorial) {**

**Intent i = new Intent(this, App\_intro.class);**

**startActivity(i);**

**}**

**DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer\_layout);**

**drawer.closeDrawer(GravityCompat.START);**

**return true;**

**}**

**}**

**Launcher.java**

**super.onCreate(savedInstanceState);**

**final String PREFS\_NAME = "Kym\_App\_Preferences";**

**SharedPreferences preferences = getSharedPreferences(PREFS\_NAME, 0);**

**if (preferences.getBoolean("logged\_in", false)) {**

**Intent i2 = new Intent(Launcher.this, MainActivity.class);**

**startActivity(i2);**

**finish();**

**}**

**else {**

**Intent i2 = new Intent(Launcher.this, Sign\_up.class);**

**startActivity(i2);**

**finish();**

**}**

**}**

**}**

**KymSMSService.java**

**package in.skylinelabs.kym;**

**import android.app.Service;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.os.IBinder;**

**import android.support.annotation.Nullable;**

**import android.util.Log;**

**import android.widget.Toast;**

**import com.google.gson.JsonObject;**

**import org.json.JSONException;**

**/\*\***

**\* Created by Jay Lohokare on 16-Oct-16.**

**\*/**

**public class KymSMSService extends Service implements AsyncTaskComplete{**

**private ActionHandler actionHandler;**

**@Override**

**public void handleResult(JsonObject result, String action) throws JSONException {**

**if (result.get("success").getAsInt() == -1) {**

**return;**

**}**

**switch (action) {**

**case "SMS":**

**if (result.get("success").getAsInt() == 1) {**

**Toast.makeText(getApplicationContext(),"Response 1", Toast.LENGTH\_SHORT).show();**

**}**

**else if(result.get("success").getAsInt() == 2){**

**Toast.makeText(getApplicationContext(),"Response 2", Toast.LENGTH\_SHORT).show();**

**}**

**else {**

**Toast.makeText(getApplicationContext(),"Amount is "+ result.get("success").getAsInt(), Toast.LENGTH\_SHORT).show();**

**}**

**break;**

**}**

**}**

**@Nullable**

**@Override**

**public IBinder onBind(Intent intent) {**

**return null;**

**}**

**@Override**

**public int onStartCommand(Intent intent, int flags, int startId) {**

**super.onStartCommand(intent, flags, startId);**

**Toast.makeText(getApplicationContext(),"Got to the Service", Toast.LENGTH\_SHORT).show();**

**Log.i("In service", "In service");**

**actionHandler = new ActionHandler(this,this);**

**String bankname = "";**

**String card = "";**

**String account\_no = "";**

**String vendor = "";**

**String amount = "";**

**String balance = "";**

**String month = "";**

**String year = "";**

**int p\_d = -1;**

**final String PREFS\_NAME = "Kym\_App\_Preferences";**

**SharedPreferences preferences = getSharedPreferences(PREFS\_NAME, 0);**

**String username = preferences.getString("name","null");**

**String type = intent.getStringExtra("TYPE");**

**if(type.equals("credit") || type.equals("debit")) {**

**bankname = intent.getStringExtra("BANK");**

**card = intent.getStringExtra("CARD");**

**vendor = intent.getStringExtra("VENDOR");**

**amount = intent.getStringExtra("AMOUNT");**

**balance = intent.getStringExtra("BALANCE");**

**month = intent.getStringExtra("MONTH");**

**year = intent.getStringExtra("YEAR");**

**actionHandler.sendSMS(bankname, card, vendor, amount, balance, month, year, type, username);**

**}**

**if(type.equals("EMI")) {**

**account\_no = intent.getStringExtra("ACCOUNT\_NO");**

**amount = intent.getStringExtra("EMI\_AMOUNT");**

**p\_d = intent.getIntExtra("P\_D", -1);**

**actionHandler.sendLoanSMS(type, account\_no, amount, p\_d, username);**

**}**

**return START\_NOT\_STICKY;**

**}**

**}**

**IncomingSMS.java**

**package in.skylinelabs.kym;**

**import android.app.ActivityManager;**

**import android.content.BroadcastReceiver;**

**import android.content.Context;**

**import android.content.Intent;**

**import android.content.SharedPreferences;**

**import android.os.Bundle;**

**import android.telephony.SmsManager;**

**import android.telephony.SmsMessage;**

**import android.util.Log;**

**import android.widget.Toast;**

**import java.text.DateFormat;**

**import java.text.SimpleDateFormat;**

**import java.util.Date;**

**/\*\***

**\* Created by Jay Lohokare on 16-Oct-16.**

**\*/**

**public class IncomingSMS extends BroadcastReceiver {**

**// Get the object of SmsManager**

**final SmsManager sms = SmsManager.getDefault();**

**private ActionHandler actionHandler;**

**String bankname;**

**String card;**

**String vendor;**

**String date;**

**String amount;**

**String balance;**

**String type;**

**String month;**

**String year;**

**int p\_d;**

**public void onReceive(Context context, Intent intent) {**

**/\* Your SBI account XXX has been credited/ddebited with Rs 100 . Current balance: 56 \*/**

**/\* Car loan EMI payment of Rs 1000 for account XXX is due.\*/**

**// Retrieves a map of extended data from the intent.**

**final Bundle bundle = intent.getExtras();**

**try {**

**//actionHandler = new ActionHandler(context, intent.);**

**//Toast.makeText(context,"In SMS recieveer", Toast.LENGTH\_SHORT).show();**

**if (bundle != null) {**

**String message = null;**

**final Object[] pdusObj = (Object[]) bundle.get("pdus");**

**for (int j = 0; j < pdusObj.length; j++) {**

**SmsMessage currentMessage = SmsMessage.createFromPdu((byte[]) pdusObj[j]);**

**String phoneNumber = currentMessage.getDisplayOriginatingAddress();**

**String senderNum = phoneNumber;**

**message = currentMessage.getDisplayMessageBody();**

**//Log.i("SmsReceiver", "senderNum: " + senderNum + "; message: " + message);**

**// Show Alert**

**/\*int duration = Toast.LENGTH\_LONG;**

**Toast toast = Toast.makeText(context,**

**"senderNum: " + senderNum + ", message: " + message, duration);**

**toast.show();\*/**

**//String msg = "Thank you for using your SBI card 7798720001 for a transaction of Rs 99 at flipkart account balance: 2000";**

**String[] tokens = message.split(" ");**

**for (String token : tokens) {**

**Log.d("token", token);**

**}**

**Date d = new Date( );**

**SimpleDateFormat mft = new SimpleDateFormat("MM");**

**SimpleDateFormat yft = new SimpleDateFormat("yyyy");**

**month = mft.format(d);**

**year = yft.format(d);**

**bankname = null;**

**card = null;**

**vendor = null;**

**amount= null;**

**balance = null;**

**p\_d = -1;**

**type = "null";**

**String keys[] = { "Rs", "at", "balance:", "account"};**

**String banknames[] = {"Barclays", "SBI", "ICICI", "HDFC"};**

**String types[] = {"credited", "debited", "EMI"};**

**String EMI\_checks[] = {"received" , "due"};**

**for (String t : types)**

**{**

**for (String token : tokens) {**

**if (token.equals(t)) {**

**type = t;**

**break;**

**}**

**}**

**}**

**if(type.equals("EMI")) {**

**for (String token : tokens)**

**{**

**if(token.equals("received")) {**

**p\_d = 1;**

**break;**

**}**

**if(token.equals("due")) {**

**p\_d = 0;**

**break;**

**}**

**}**

**}**

**for (String bank : banknames)**

**{**

**for (String token : tokens) {**

**if (token.equals(bank)) {**

**bankname = bank;**

**break;**

**}**

**}**

**}**

**for(int i = 0; i < tokens.length; i++) {**

**if(tokens[i].equals(keys[0])) {**

**amount = tokens[i + 1];**

**}**

**if(tokens[i].equals(keys[1])) {**

**vendor = tokens[i + 1];**

**}**

**if(tokens[i].equals(keys[2])) {**

**balance = tokens[i + 1];**

**}**

**if(tokens[i].equals(keys[3])) {**

**card = tokens[i + 1];**

**}**

**}**

**}**

**if(type.equals("credited"))**

**type = "credit";**

**if(type.equals("debited"))**

**type = "debit";**

**if(!type.equals("null")) {**

**Toast.makeText(context,"Type got", Toast.LENGTH\_SHORT).show();**

**}**

**/\*System.out.println(("bankname" + bankname));**

**System.out.println("card" + card);**

**System.out.println("vendor" + vendor);**

**System.out.println("amount" + amount);**

**System.out.println("balance" + balance);\*/**

**if(type.equals("credit") || type.equals("debit")) {**

**Intent i = new Intent(context, KymSMSService.class);**

**i.putExtra("BANK", bankname);**

**i.putExtra("CARD", card);**

**i.putExtra("VENDOR", vendor);**

**i.putExtra("AMOUNT", amount);**

**i.putExtra("BALANCE", balance);**

**i.putExtra("MONTH", month);**

**i.putExtra("YEAR", year);**

**i.putExtra("TYPE", type);**

**context.startService(i);**

**}**

**if(type.equals("EMI")) {**

**Intent i = new Intent(context, KymSMSService.class);**

**i.putExtra("ACCOUNT\_NO", card);**

**i.putExtra("EMI\_AMOUNT", amount);**

**i.putExtra("TYPE", type);**

**i.putExtra("P\_D", p\_d);**

**context.startService(i);**

**}**

**}**

**} catch (Exception e) {**

**Log.e("SmsReceiver", "Exception smsReceiver" + e);**

**}**

**}**

**}**

**HTTPJSONPost.java**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.io.OutputStreamWriter;**

**import java.io.StringReader;**

**import java.net.HttpURLConnection;**

**import java.net.ProtocolException;**

**import java.net.URL;**

**public class HttpJsonPost extends AsyncTask<JsonObject, Integer, JsonObject> {**

**private String url, progressinfo, action;**

**private Context context;**

**private ProgressDialog progressDialog;**

**private AsyncTaskComplete callback;**

**private boolean progressbar;**

**public HttpJsonPost(String url, String action, String progressinfo, Context context, AsyncTaskComplete callback) {**

**this.context = context;**

**this.url = url;**

**this.callback = callback;**

**this.action = action;**

**this.progressbar = !progressinfo.isEmpty();**

**this.progressinfo = progressinfo;**

**}**

**public void setProgressinfo(String progressinfo) {**

**this.progressinfo = progressinfo;**

**}**

**@Override**

**protected void onPreExecute() {**

**if (progressbar && action=="Login") {**

**progressDialog = new ProgressDialog(context);**

**progressDialog.setMessage(progressinfo);**

**progressDialog.setCancelable(false);**

**progressDialog.show();**

**}**

**}**

**@Override**

**protected JsonObject doInBackground(JsonObject... params) {**

**return postJsonObject(params[0]);**

**}**

**@Override**

**protected void onPostExecute(JsonObject result) {**

**if (progressbar && action=="Login") {**

**progressDialog.dismiss();**

**}**

**try {**

**callback.handleResult(result, action);**

**} catch (JSONException e) {**

**e.printStackTrace();**

**}**

**}**

**private JsonObject postJsonObject(JsonObject jsonObject) {**

**JsonObject nullresult = new JsonObject();**

**nullresult.addProperty("success", -1);**

**try {**

**String data = jsonObject.toString();**

**URL object = new URL(url);**

**HttpURLConnection httpURLConnection = (HttpURLConnection) object.openConnection();**

**httpURLConnection.setDoOutput(true);**

**httpURLConnection.setDoInput(true);**

**httpURLConnection.setRequestProperty("Content-Type", "application/json");**

**httpURLConnection.setRequestProperty("Accept", "application/json");**

**httpURLConnection.setRequestMethod("POST");**

**httpURLConnection.setConnectTimeout(4000);**

**httpURLConnection.setReadTimeout(15000);**

**OutputStreamWriter wr = new OutputStreamWriter(httpURLConnection.getOutputStream());**

**wr.write(data);**

**wr.flush();**

**StringBuilder sb = new StringBuilder();**

**int HttpResult = httpURLConnection.getResponseCode();**

**if (HttpResult == HttpURLConnection.HTTP\_OK) {**

**BufferedReader br = new BufferedReader(**

**new InputStreamReader(httpURLConnection.getInputStream(), "utf-8"));**

**String line;**

**while ((line = br.readLine()) != null) {**

**sb.append(line).append("\n");**

**}**

**br.close();**

**}**

**JsonObject result = new JsonParser().parse(sb.toString()).getAsJsonObject();**

**Log.i("This is the result" + result, "This is the result" + result);**

**if (result != null)**

**return result;**

**else**

**return nullresult;**

**} catch (ProtocolException e) {**

**e.printStackTrace();**

**} catch (MalformedJsonException e) {**

**e.printStackTrace();**

**} catch (IOException e) {**

**e.printStackTrace();**

**}**

**return nullresult;**

**}**

**}**

**DemoPageAdapter.java**

**public DemoPagerAdapter(FragmentManager fm) {**

**super(fm);**

**}**

**@Override**

**public Fragment getItem(int i) {**

**if(i==0)**

**return ColorFragment.newInstance(Color.parseColor("#0277bd"), i);//red**

**if(i==1)**

**return ColorFragment.newInstance(Color.parseColor("#4baf4f"), i);//pink**

**if(i==2)**

**return ColorFragment.newInstance(Color.parseColor("#f29d0f"), i);//green**

**if(i==3)**

**return ColorFragment.newInstance(Color.parseColor("#5f7d88"), i);**

**if(i==4)**

**return ColorFragment.newInstance(Color.parseColor("#0277bd"), i);**

**else**

**return null;**

**}**

**@Override**

**public int getCount() {**

**return pagerCount;**

**}**

**}**

**DatabaseHandler.java**

**/\*\***

**\* Created by Reshul Dani on 04-09-2015.**

**\*/**

**public class DatabaseHandler extends SQLiteOpenHelper {**

**private static final int DATABASE\_VERSION = 1;**

**// Database Name**

**private static final String DATABASE\_NAME = "kymMessages";**

**// Contacts table name**

**private static final String TABLE = "messages";**

**// Contacts Table Columns names**

**private static final String KEY\_ID = "id";**

**private static final String KEY\_DATE = "date";**

**private static final String KEY\_MESSAGE = "message";**

**private static final String KEY\_TAG = "tag";**

**public DatabaseHandler(Context context) {**

**super(context, DATABASE\_NAME, null, DATABASE\_VERSION);**

**}**

**@Override**

**public void onCreate(SQLiteDatabase db) {**

**String CREATE\_TABLE = "CREATE TABLE " + TABLE + "("**

**+ KEY\_ID + " INTEGER PRIMARY KEY," + KEY\_DATE + " TEXT," + KEY\_MESSAGE + " TEXT,"**

**+ KEY\_TAG + " TEXT" + ")";**

**db.execSQL(CREATE\_TABLE);**

**}**

**@Override**

**public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {**

**db.execSQL("DROP TABLE IF EXISTS " + TABLE);**

**// Create tables again**

**onCreate(db);**

**}**

**void addtodatabase(ChatMessage cm) {**

**SQLiteDatabase db = this.getWritableDatabase();**

**ContentValues values = new ContentValues();**

**values.put(KEY\_DATE, cm.getDate());**

**values.put(KEY\_MESSAGE, cm.getMessage());**

**values.put(KEY\_TAG, cm.getTag());**

**// Inserting Row**

**db.insert(TABLE, null, values);**

**db.close(); // Closing database connection**

**}**

**public List<ChatMessage> getAllMessages() {**

**List<ChatMessage> messageList = new ArrayList<ChatMessage>();**

**// Select All Query**

**String selectQuery = "SELECT \* FROM " + TABLE;**

**SQLiteDatabase db = this.getWritableDatabase();**

**Cursor cursor = db.rawQuery(selectQuery, null);**

**// looping through all rows and adding to list**

**if (cursor.moveToFirst()) {**

**do {**

**ChatMessage cm = new ChatMessage();**

**cm.setId(Integer.parseInt(cursor.getString(0)));**

**cm.setDate(cursor.getString(1));**

**cm.setMessage(cursor.getString(2));**

**cm.setTag(Integer.parseInt(cursor.getString(3)));**

**// Adding contact to list**

**messageList.add(cm);**

**} while (cursor.moveToNext());**

**}**

**return messageList;**

**}**

**}**

**ColorFragment.java**

**package in.skylinelabs.kym;**

**import android.os.Bundle;**

**import android.support.v4.app.Fragment;**

**import android.view.LayoutInflater;**

**import android.view.View;**

**import android.view.ViewGroup;**

**public class ColorFragment extends Fragment {**

**private static final String ARG\_COLOR = "color";**

**private static final String ARG\_INT = "position";**

**private int mColor, pos;**

**public static ColorFragment newInstance(int param1, int param2) {**

**ColorFragment fragment = new ColorFragment();**

**Bundle args = new Bundle();**

**args.putInt(ARG\_COLOR, param1);**

**args.putInt(ARG\_INT, param2);**

**fragment.setArguments(args);**

**return fragment;**

**}**

**public ColorFragment() {**

**}**

**@Override**

**public void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**if (getArguments() != null) {**

**mColor = getArguments().getInt(ARG\_COLOR);**

**pos = getArguments().getInt(ARG\_INT);**

**}**

**}**

**@Override**

**public View onCreateView(LayoutInflater inflater, ViewGroup container,**

**Bundle savedInstanceState) {**

**if (pos == 0) {**

**View v = inflater.inflate(R.layout.intro\_1, container, false);**

**v.setBackgroundColor(mColor);**

**return v;**

**}**

**if (pos == 1) {**

**View v = inflater.inflate(R.layout.intro\_2, container, false);**

**v.setBackgroundColor(mColor);**

**return v;**

**}**

**if (pos == 2) {**

**View v = inflater.inflate(R.layout.intro\_3, container, false);**

**v.setBackgroundColor(mColor);**

**return v;**

**}**

**if (pos == 3) {**

**View v = inflater.inflate(R.layout.intro\_4, container, false);**

**v.setBackgroundColor(mColor);**

**return v;**

**}**

**if (pos == 4) {**

**View v = inflater.inflate(R.layout.intro\_5, container, false);**

**v.setBackgroundColor(mColor);**

**return v;**

**}**

**else**

**return null;**

**}**

**}**

**ChatMessage.java**

**public long getId() {**

**return id;**

**}**

**public void setId(long id) {**

**this.id = id;**

**}**

**public int getTag() {**

**return tag;**

**}**

**public void setTag(int isMe) {**

**this.tag = isMe;**

**}**

**public String getMessage() {**

**return message;**

**}**

**public void setMessage(String message) {**

**this.message = message;**

**}**

**public String getDate() {**

**return dateTime;**

**}**

**public void setDate(String dateTime) {**

**this.dateTime = dateTime;**

**}**

**}**

**ChatAdapter.java**

**import java.util.List;**

**/\*\***

**\* Created by Technovibe on 17-04-2015.**

**\*/**

**public class ChatAdapter extends BaseAdapter {**

**private final List<ChatMessage> chatMessages;**

**private Activity context;**

**public ChatAdapter(Activity context, List<ChatMessage> chatMessages) {**

**this.context = context;**

**this.chatMessages = chatMessages;**

**}**

**@Override**

**public int getCount() {**

**if (chatMessages != null) {**

**return chatMessages.size();**

**} else {**

**return 0;**

**}**

**}**

**@Override**

**public ChatMessage getItem(int position) {**

**if (chatMessages != null) {**

**return chatMessages.get(position);**

**} else {**

**return null;**

**}**

**}**

**@Override**

**public long getItemId(int position) {**

**return position;**

**}**

**@Override**

**public View getView(final int position, View convertView, ViewGroup parent) {**

**ViewHolder holder;**

**ChatMessage chatMessage = getItem(position);**

**LayoutInflater vi = (LayoutInflater) context.getSystemService(Context.LAYOUT\_INFLATER\_SERVICE);**

**if (convertView == null) {**

**convertView = vi.inflate(R.layout.list\_item\_chat\_message, null);**

**holder = createViewHolder(convertView);**

**convertView.setTag(holder);**

**} else {**

**holder = (ViewHolder) convertView.getTag();**

**}**

**int myMsg = chatMessage.getTag() ;//Just a dummy check to simulate whether it me or other sender**

**setAlignment(holder, myMsg);**

**holder.txtMessage.setText(chatMessage.getMessage());**

**holder.txtInfo.setText(chatMessage.getDate());**

**return convertView;**

**}**

**public void add(ChatMessage message) {**

**chatMessages.add(message);**

**}**

**public void add(List<ChatMessage> messages) {**

**chatMessages.addAll(messages);**

**}**

**private void setAlignment(ViewHolder holder, int isMe) {**

**if (isMe != 0) {**

**holder.contentWithBG.setBackgroundResource(R.drawable.untitled\_4);**

**LinearLayout.LayoutParams layoutParams = (LinearLayout.LayoutParams) holder.contentWithBG.getLayoutParams();**

**layoutParams.gravity = Gravity.RIGHT;**

**holder.contentWithBG.setLayoutParams(layoutParams);**

**RelativeLayout.LayoutParams lp = (RelativeLayout.LayoutParams) holder.content.getLayoutParams();**

**lp.addRule(RelativeLayout.ALIGN\_PARENT\_LEFT, 0);**

**lp.addRule(RelativeLayout.ALIGN\_PARENT\_RIGHT);**

**holder.content.setLayoutParams(lp);**

**layoutParams = (LinearLayout.LayoutParams) holder.txtMessage.getLayoutParams();**

**layoutParams.gravity = Gravity.RIGHT;**

**holder.txtMessage.setLayoutParams(layoutParams);**

**layoutParams = (LinearLayout.LayoutParams) holder.txtInfo.getLayoutParams();**

**layoutParams.gravity = Gravity.RIGHT;**

**holder.txtInfo.setLayoutParams(layoutParams);**

**} else {**

**holder.contentWithBG.setBackgroundResource(R.drawable.untitled\_3);**

**LinearLayout.LayoutParams layoutParams = (LinearLayout.LayoutParams) holder.contentWithBG.getLayoutParams();**

**layoutParams.gravity = Gravity.LEFT;**

**holder.contentWithBG.setLayoutParams(layoutParams);**

**RelativeLayout.LayoutParams lp = (RelativeLayout.LayoutParams) holder.content.getLayoutParams();**

**lp.addRule(RelativeLayout.ALIGN\_PARENT\_RIGHT, 0);**

**lp.addRule(RelativeLayout.ALIGN\_PARENT\_LEFT);**

**holder.content.setLayoutParams(lp);**

**layoutParams = (LinearLayout.LayoutParams) holder.txtMessage.getLayoutParams();**

**layoutParams.gravity = Gravity.LEFT;**

**holder.txtMessage.setLayoutParams(layoutParams);**

**layoutParams = (LinearLayout.LayoutParams) holder.txtInfo.getLayoutParams();**

**layoutParams.gravity = Gravity.LEFT;**

**holder.txtInfo.setLayoutParams(layoutParams);**

**}**

**}**

**private ViewHolder createViewHolder(View v) {**

**ViewHolder holder = new ViewHolder();**

**holder.txtMessage = (TextView) v.findViewById(R.id.txtMessage);**

**holder.content = (LinearLayout) v.findViewById(R.id.content);**

**holder.contentWithBG = (LinearLayout) v.findViewById(R.id.contentWithBackground);**

**holder.txtInfo = (TextView) v.findViewById(R.id.txtInfo);**

**return holder;**

**}**

**private static class ViewHolder {**

**public TextView txtMessage;**

**public TextView txtInfo;**

**public LinearLayout content;**

**public LinearLayout contentWithBG;**

**}**

**}**

**ChatActivity.java**

**import android.text.TextUtils;**

**import android.util.Log;**

**import android.view.Menu;**

**import android.view.MenuItem;**

**import android.view.View;**

**import android.widget.Button;**

**import android.widget.EditText;**

**import android.widget.ImageView;**

**import android.widget.ListView;**

**import android.widget.RelativeLayout;**

**import android.widget.TextView;**

**import android.widget.Toast;**

**import com.google.gson.JsonObject;**

**import org.json.JSONException;**

**import java.text.DateFormat;**

**import java.text.SimpleDateFormat;**

**import java.util.ArrayList;**

**import java.util.Date;**

**import java.util.List;**

**public class ChatActivity extends ActionBarActivity implements AsyncTaskComplete {**

**private EditText messageET, edt;**

**private ListView messagesContainer;**

**private ImageView sendBtn;**

**private ChatAdapter adapter;**

**private ArrayList<ChatMessage> chatHistory;**

**private DatabaseHandler db;**

**protected static final int RESULT\_SPEECH = 1;**

**private int account\_flag = 0;**

**String year, month, username;**

**private ActionHandler actionHandler;**

**private static final String [] DANGEROUS\_PERMISSIONS = {**

**Manifest.permission.READ\_PHONE\_STATE,**

**Manifest.permission.READ\_SMS,**

**Manifest.permission.READ\_EXTERNAL\_STORAGE**

**};**

**private void initPermissions() {**

**List<String> missingPermissions = new ArrayList<String>();**

**for(String permission : DANGEROUS\_PERMISSIONS) {**

**if(ActivityCompat.checkSelfPermission(this, permission) != PackageManager.PERMISSION\_GRANTED) {**

**missingPermissions.add(permission);**

**}**

**}**

**if (missingPermissions.size() > 0) {**

**String [] permissions = new String[missingPermissions.size()];**

**ActivityCompat.requestPermissions(**

**this,**

**missingPermissions.toArray(permissions),**

**1);**

**} else {**

**// we have all permissions, move on**

**}**

**}**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_chat\_main);**

**actionHandler = new ActionHandler(this, ChatActivity.this);**

**Date d = new Date( );**

**SimpleDateFormat mft = new SimpleDateFormat("MM");**

**SimpleDateFormat yft = new SimpleDateFormat("yyyy");**

**month = mft.format(d);**

**year = yft.format(d);**

**final String PREFS\_NAME = "Kym\_App\_Preferences";**

**SharedPreferences preferences = getSharedPreferences(PREFS\_NAME, 0);**

**username = preferences.getString("name","null");**

**initPermissions();**

**initControls();**

**ImageView mic = (ImageView) findViewById(R.id.micButton);**

**mic.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**Intent intent = new Intent(**

**RecognizerIntent.ACTION\_RECOGNIZE\_SPEECH);**

**intent.putExtra(RecognizerIntent.EXTRA\_LANGUAGE\_MODEL, "en-US");**

**try {**

**startActivityForResult(intent, RESULT\_SPEECH);**

**edt = (EditText) findViewById(R.id.messageEdit);**

**edt.setText("");**

**} catch (ActivityNotFoundException a) {**

**Toast t = Toast.makeText(getApplicationContext(),**

**"Ops! Your device doesn't support Speech to Text",**

**Toast.LENGTH\_SHORT);**

**t.show();**

**}**

**}**

**});**

**loadHistory();**

**DisplayContent("Hey! Kym here. \nHere is what I know about you till now:");**

**actionHandler.getUserSummary(username);**

**}**

**@Override**

**public boolean onCreateOptionsMenu(Menu menu) {**

**return true;**

**}**

**@Override**

**public boolean onOptionsItemSelected(MenuItem item) {**

**return super.onOptionsItemSelected(item);**

**}**

**private void initControls() {**

**db = new DatabaseHandler(this);**

**messagesContainer = (ListView) findViewById(R.id.messagesContainer);**

**messageET = (EditText) findViewById(R.id.messageEdit);**

**sendBtn = (ImageView) findViewById(R.id.sendButton);**

**sendBtn.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**String messageText = messageET.getText().toString();**

**if (TextUtils.isEmpty(messageText)) {**

**return;**

**}**

**ChatMessage chatMessage = new ChatMessage();**

**chatMessage.setId(122);//dummy**

**chatMessage.setMessage(messageText);**

**chatMessage.setDate(DateFormat.getDateTimeInstance().format(new Date()));**

**chatMessage.setTag(1);**

**messageET.setText("");**

**Log.d("Insert: ", "Inserting ..");**

**db.addtodatabase(chatMessage);**

**displayMessage(chatMessage);**

**String[] tokens = messageText.split(" ");**

**String account = "";**

**String keys[] = {"balance"};**

**String main\_key = "";**

**int i = 0;**

**for (String token : tokens) {**

**if(token.matches("account")) {**

**account = tokens[i + 1] ;**

**account\_flag = 1;**

**}**

**for (String key : keys) {**

**if (token.matches(key)) {**

**main\_key = key;**

**break;**

**}**

**}**

**i++;**

**}**

**if (main\_key.matches("balance") || account\_flag == 1) {**

**Toast.makeText(getApplicationContext(),"asked for balance", Toast.LENGTH\_LONG).show();**

**if(account\_flag == 1){**

**Log.i("account no", account);**

**Toast.makeText(getApplicationContext(),"account number present", Toast.LENGTH\_LONG).show();**

**actionHandler.get\_account\_balance(username, tokens[0], month, year);**

**account\_flag = 0;**

**}**

**else {**

**Toast.makeText(getApplicationContext(),"account number not there", Toast.LENGTH\_LONG).show();**

**actionHandler.get\_accounts(username);**

**account\_flag = 1;**

**}**

**}**

**}**

**});**

**adapter = new ChatAdapter(ChatActivity.this, new ArrayList<ChatMessage>());**

**messagesContainer.setAdapter(adapter);**

**}**

**public void displayMessage(ChatMessage message) {**

**adapter.add(message);**

**adapter.notifyDataSetChanged();**

**scroll();**

**}**

**public void DisplayContent(String message1)**

**{**

**ChatMessage m = new ChatMessage();**

**m.setMessage(message1);**

**m.setDate(DateFormat.getDateTimeInstance().format(new Date()));**

**m.setTag(0);**

**db.addtodatabase(m);**

**displayMessage(m);**

**}**

**private void scroll() {**

**messagesContainer.setSelection(messagesContainer.getCount() - 1);**

**}**

**@Override**

**protected void onActivityResult(int requestCode, int resultCode, Intent data) {**

**super.onActivityResult(requestCode, resultCode, data);**

**switch (requestCode) {**

**case RESULT\_SPEECH: {**

**if (resultCode == RESULT\_OK && null != data) {**

**ArrayList<String> text = data**

**.getStringArrayListExtra(RecognizerIntent.EXTRA\_RESULTS);**

**edt.setText(text.get(0));**

**}**

**break;**

**}**

**}**

**}**

**private void loadHistory(){**

**adapter = new ChatAdapter(ChatActivity.this, new ArrayList<ChatMessage>());**

**messagesContainer.setAdapter(adapter);**

**Log.d("Reading: ", "Reading all contacts..");**

**List<ChatMessage> cm = db.getAllMessages();**

**for (ChatMessage cn : cm) {**

**String log = "Id: " + cn.getId() + " ,Date: " + cn.getDate() + " ,Message: " + cn.getMessage() + " ,Tag: " + cn.getTag();**

**Log.d("Row: ", log);**

**}**

**for(int i=0; i<cm.size(); i++) {**

**ChatMessage message = cm.get(i);**

**displayMessage(message);**

**}**

**}**

**@Override**

**public void handleResult(JsonObject result, String action) throws JSONException {**

**if (result.get("success").getAsInt() == -1) {**

**return;**

**}**

**switch (action) {**

**case "Account\_balance":**

**DisplayContent(result.get("msg").getAsString());**

**break;**

**case "get\_accounts" :**

**DisplayContent(result.get("msg").getAsString());**

**break;**

**case "Get\_User\_Summary":**

**DisplayContent(result.get("msg").getAsString());**

**break;**

**}**

**}**

**}**

**AsyncTaskComplete.java**

**package in.skylinelabs.kym;**

**import com.google.gson.JsonObject;**

**import org.json.JSONException;**

**/\*\***

**\* Created by Jay Lohokare on 16-Oct-16.**

**\*/**

**public interface AsyncTaskComplete {**

**void handleResult(JsonObject result, String action) throws JSONException;**

**}**

**App\_intro.java**

**package in.skylinelabs.kym;**

**import android.content.Intent;**

**import android.graphics.Color;**

**import android.os.Build;**

**import android.os.Bundle;**

**import android.support.v4.view.ViewPager;**

**import android.support.v7.app.ActionBarActivity;**

**import android.view.View;**

**import android.widget.TextView;**

**import me.relex.circleindicator.CircleIndicator;**

**import me.relex.circleindicator.CircleIndicator;**

**public class App\_intro extends ActionBarActivity {**

**TextView skip, next;**

**Boolean is\_called\_from\_login;**

**@Override protected void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_app\_intro);**

**is\_called\_from\_login = getIntent().getBooleanExtra("is\_called\_from\_login",false);**

**getWindow().getDecorView().setSystemUiVisibility(**

**View.SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE**

**| View.SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN);**

**skip = (TextView) findViewById(R.id.skip);**

**next = (TextView) findViewById(R.id.next);**

**skip.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**onBackPressed();**

**}**

**});**

**if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.LOLLIPOP) {**

**getWindow().setStatusBarColor(Color.TRANSPARENT);**

**}**

**final ViewPager customViewpager = (ViewPager) findViewById(R.id.viewpager\_custom);**

**CircleIndicator customIndicator = (CircleIndicator) findViewById(R.id.indicator\_custom);**

**DemoPagerAdapter customPagerAdapter = new DemoPagerAdapter(getSupportFragmentManager());**

**customViewpager.setAdapter(customPagerAdapter);**

**customIndicator.setViewPager(customViewpager);**

**customViewpager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {**

**@Override**

**public void onPageScrolled(int i, float v, int i2) {**

**if(customViewpager.getCurrentItem()==4) {**

**next.setText("Done");**

**next.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**onBackPressed();**

**}**

**});**

**}**

**}**

**@Override**

**public void onPageSelected(int i) {**

**}**

**@Override**

**public void onPageScrollStateChanged(int i) {**

**}**

**});**

**if(customViewpager.getCurrentItem()!=5) {**

**next.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**customViewpager.setCurrentItem(customViewpager.getCurrentItem() + 1);**

**}**

**});**

**}**

**customViewpager.setPageTransformer(true, new ViewPager.PageTransformer() {**

**@Override**

**public void transformPage(View view, float position) {**

**// Ensures the views overlap each other.**

**view.setTranslationX(view.getWidth() \* -position);**

**// Alpha property is based on the view position.**

**if (position <= -1.0F || position >= 1.0F) {**

**view.setAlpha(0.0F);**

**} else if (position == 0.0F) {**

**view.setAlpha(1.0F);**

**} else { // position is between -1.0F & 0.0F OR 0.0F & 1.0F**

**view.setAlpha(1.0F - Math.abs(position));**

**}**

**}**

**});**

**}**

**@Override**

**public void onBackPressed(){**

**if(is\_called\_from\_login){**

**startActivity(new Intent(this,MainActivity.class));**

**finish();**

**}**

**else{**

**finish();**

**}**

**}**

**}**

**ActionHandler.java**

**package in.skylinelabs.kym;**

**import android.app.ProgressDialog;**

**import android.content.Context;**

**import android.content.SharedPreferences;**

**import android.util.Log;**

**import com.google.gson.JsonArray;**

**import com.google.gson.JsonObject;**

**import java.text.SimpleDateFormat;**

**import java.util.Date;**

**import java.util.HashMap;**

**public class ActionHandler {**

**AsyncTaskComplete callback;**

**ProgressDialog progressDialog;**

**private Context context;**

**private String address = "192.168.43.128:80";**

**private String ACCOUNT\_URL = "http://" + address +"/user.php";**

**private String SMS\_URL = "http://" + address +"/sms.php";**

**private String LOAN\_SMS\_URL = "http://" + address +"/loan\_sms.php";**

**private String CHAT\_BOT\_URL = "http://" + address +"/summary.php";**

**private String ACCOUNT\_BALANCE\_URL = "http://" + address +"/give\_balance.php";**

**private String GET\_ACCOUNTS\_URL = "http://" + address +"/give\_account.php";**

**public ActionHandler(Context context, AsyncTaskComplete callback) {**

**this.callback = callback;**

**this.context = context;**

**progressDialog = new ProgressDialog(context);**

**}**

**public void login(String name, String twitter\_handle ) {**

**postJsonObject(createLoginJsonObject(name, twitter\_handle), ACCOUNT\_URL, "Login", "Logging In\nPlease Wait");**

**}**

**public void sendSMS(String bankname, String card, String vendor, String amount, String balance, String month, String year, String type, String username) {**

**postJsonObject(createSMSJsonObject(bankname, card, vendor, amount, balance, month, year, type, username), SMS\_URL, "SMS", "Logging In\nPlease Wait");**

**}**

**public void sendLoanSMS(String type, String account\_no, String amount, int p\_d, String username) {**

**postJsonObject(createLoanSMSJsonObject(type, account\_no, amount, p\_d, username), LOAN\_SMS\_URL, "LOAN\_SMS", "Logging In\nPlease Wait");**

**}**

**public void getUserSummary(String username) {**

**postJsonObject(createGetUserSummaryJsonObject(username), CHAT\_BOT\_URL, "Get\_User\_Summary", "Logging In\nPlease Wait");**

**}**

**public void get\_account\_balance(String username,String account, String month, String year) {**

**JsonObject jsonObject = new JsonObject();**

**jsonObject.addProperty("username", username);**

**jsonObject.addProperty("account\_number", account);**

**jsonObject.addProperty("month", month);**

**jsonObject.addProperty("year", year);**

**postJsonObject(jsonObject, ACCOUNT\_BALANCE\_URL, "Account\_balance", "Getting your balance\nPlease wait");**

**}**

**public void get\_accounts(String username){**

**JsonObject jsonObject = new JsonObject();**

**jsonObject.addProperty("username", username);**

**postJsonObject(jsonObject, GET\_ACCOUNTS\_URL, "get\_accounts", "Getting your accounts\nPlease wait");**

**}**

**private JsonObject createLoginJsonObject(String name, String twitter\_handle) {**

**JsonObject jsonObject = new JsonObject();**

**jsonObject.addProperty("username", name);**

**jsonObject.addProperty("twitter\_handle", twitter\_handle);**

**return jsonObject;**

**}**

**private JsonObject createSMSJsonObject(String bank\_name, String card, String vendor, String amount, String balance, String month, String year, String type, String username) {**

**JsonObject jsonObject = new JsonObject();**

**jsonObject.addProperty("bank\_name", bank\_name);**

**jsonObject.addProperty("account\_number", card);**

**jsonObject.addProperty("vendor", vendor);**

**jsonObject.addProperty("amount", amount);**

**jsonObject.addProperty("balance", balance);**

**jsonObject.addProperty("month", month);**

**jsonObject.addProperty("year", year);**

**jsonObject.addProperty("type", type);**

**jsonObject.addProperty("username", username);**

**return jsonObject;**

**}**

**private JsonObject createLoanSMSJsonObject(String type, String account\_no, String amount, int p\_d, String username) {**

**JsonObject jsonObject = new JsonObject();**

**jsonObject.addProperty("account\_number", account\_no);**

**jsonObject.addProperty("amount", amount);**

**jsonObject.addProperty("type", type);**

**jsonObject.addProperty("username", username);**

**jsonObject.addProperty("p\_d", p\_d);**

**Log.i("c s", type);**

**Log.i(" c s", account\_no);**

**Log.i("scv s", amount);**

**Log.i("sv",username);**

**Log.i("ssdv", String.valueOf(p\_d));**

**Log.i("name", "rshl");**

**return jsonObject;**

**}**

**private JsonObject createGetUserSummaryJsonObject(String name) {**

**JsonObject jsonObject = new JsonObject();**

**Date d = new Date( );**

**SimpleDateFormat mft = new SimpleDateFormat("MM");**

**SimpleDateFormat yft = new SimpleDateFormat("yyyy");**

**String month = mft.format(d);**

**String year = yft.format(d);**

**jsonObject.addProperty("username", name);**

**jsonObject.addProperty("month", month);**

**jsonObject.addProperty("year", year);**

**return jsonObject;**

**}**

**private void postJsonObject(final JsonObject jsonObject, String url, final String action, String progress\_status) {**

**HttpJsonPost httpJsonPost = new HttpJsonPost(url, action, progress\_status, context, callback);**

**httpJsonPost.execute(jsonObject);**

**}**

**}**