import android.content.ActivityNotFoundException;

import android.os.AsyncTask;

import android.os.Bundle;

import java.io.BufferedInputStream;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

import java.util.UUID;

import android.annotation.SuppressLint;

import android.app.Activity;

import android.app.AlertDialog;

import android.app.ProgressDialog;

import android.bluetooth.BluetoothAdapter;

import android.bluetooth.BluetoothDevice;

import android.bluetooth.BluetoothSocket;

import android.content.Context;

import android.content.DialogInterface;

import android.content.DialogInterface.OnClickListener;

import android.content.Intent;

import android.hardware.Sensor;

import android.hardware.SensorManager;

import android.speech.RecognizerIntent;

import android.view.animation.Animation;

import android.view.animation.RotateAnimation;

import android.os.Handler;

import android.os.Message;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

public class MainActivity extends Activity {

    TextView txtv1,txtv2,txtv3,txtv4,txtLvl;

    Handler hos;

    String tag,val,pval;

    URL url1 ;

    URL url;

    ////////\*\*\*\*\*declartion part\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/////////////////////////////////////////////

    TTSManager ttsManager = null;

    protected static final int RESULT\_SPEECH = 0;

    @Override

    public void onCreate(Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        txtv1=(TextView)findViewById(R.id.txtv1);

        txtv2=(TextView)findViewById(R.id.txtv2);

        txtv3=(TextView)findViewById(R.id.txtv3);

        txtv4=(TextView)findViewById(R.id.txtv4);

        hos=new Handler();

        new Thread(new Runnable()

        {

            @Override

            public void run() {

                // TODO Auto-generated method stub

                while(true)

                {

                    try

                    {

                        Thread.sleep(10000);

                        hos.post(new Runnable()

                        {

                            @Override

                            public void run() {

                                // TODO Auto-generated method stub

                                Updatedata1();

                            }

                        });

                    }

                    catch(Exception e)

                    {

                        e.printStackTrace();

                    }

                }

            }

        }).start();

        ////////////////////\*\*\*\*\*\*\*\*\*\*merge with design form-start\*\*\*\*\*////////////////////////

        ttsManager = new TTSManager();

        ttsManager.init(this);

    }

       public void btnl1n(View view) {

        try {

            url= new URL("http://mangocity.appblocky.com/webdb/storeavalue.php?tag=nsh0408read&value=\*1");

        } catch (MalformedURLException e) {

            e.printStackTrace();

        }

        Updatedata();

    }

    public void btnl1f(View view) {

        try {

            url= new URL("http://mangocity.appblocky.com/webdb/storeavalue.php?tag=nsh0296read&value=\*2");

        } catch (MalformedURLException e) {

            e.printStackTrace();

        }

        Updatedata();

    }

    public void btnl2n(View view) {

        try {

            url= new URL("http://mangocity.appblocky.com/webdb/storeavalue.php?tag=nsh0296read&value=\*3");

        } catch (MalformedURLException e) {

            e.printStackTrace();

        }

        Updatedata();

    }

    //////////////////////////////////////////////////////////////////

    private void Updatedata(){

        if (AppStatus.getInstance(getApplicationContext()).isOnline()) {

            Toast.makeText(getApplicationContext(),"internet connected",Toast.LENGTH\_SHORT).show();

            MyAsync myAsync = new MyAsync();

            myAsync.execute();

        }

        else {

            Toast.makeText(getApplicationContext(),"internet not connected",Toast.LENGTH\_SHORT).show();

            AlertDialog.Builder builder = new AlertDialog.

                    Builder(MainActivity.this);

            builder.setTitle("Alert!");

            builder.setMessage("Please check your network connection");

            builder.setPositiveButton("OK", new OnClickListener()

            {

                @Override

                public void onClick(DialogInterface dialogInterface, int i) {

                    finish();

                }

            });

            builder.create().show();

        }

    }

    class MyAsync extends AsyncTask<Void, Void, String> {

        @Override

        protected void onPreExecute() {

            super.onPreExecute();

        }

        @Override

        protected String doInBackground(Void... voids) {

            String result = null;

            try {

                HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();

                InputStream in = new BufferedInputStream(urlConnection.getInputStream());

                result = inputStreamToString(in);

            } catch (Exception e) {

                e.printStackTrace();

            }

            return result;

        }

        @Override

        protected void onPostExecute(String s) {

            super.onPostExecute(s);

            if(s!=null) {

                }else{

                }

        }

    }

    private String inputStreamToString(InputStream is) {

        String rLine = "";

        StringBuilder answer = new StringBuilder();

        InputStreamReader isr = new InputStreamReader(is);

        BufferedReader rd = new BufferedReader(isr);

        try {

            while ((rLine = rd.readLine()) != null) {

                answer.append(rLine);

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

        return answer.toString();

    }

    private void Updatedata1(){

        if (AppStatus.getInstance(getApplicationContext()).isOnline()) {

            Toast.makeText(getApplicationContext(),"internet connected",Toast.LENGTH\_SHORT).show();

            MyAsync1 myAsync1 = new MyAsync1();

            myAsync1.execute();

        }

        else {

            Toast.makeText(getApplicationContext(),"internet not connected",Toast.LENGTH\_SHORT).show();

            AlertDialog.Builder builder = new AlertDialog.

                    Builder(MainActivity.this);

            builder.setTitle("Alert!");

            builder.setMessage("Please check your network connection");

            builder.setPositiveButton("OK", new DialogInterface.OnClickListener()

            {

                @Override

                public void onClick(DialogInterface dialogInterface, int i) {

                    finish();

                }

            });

            builder.create().show();

        }

    }

    class MyAsync1 extends AsyncTask<Void, Void, String> {

        @Override

        protected void onPreExecute() {

            super.onPreExecute();

        }

        @Override

        protected String doInBackground(Void... voids) {

            String result = null;

            try {

                                url1 = new URL("http://mangocity.appblocky.com/webdb/getvalue.php?tag=nsh0411");

                HttpURLConnection urlConnection = (HttpURLConnection) url1.openConnection();

                InputStream in = new BufferedInputStream(urlConnection.getInputStream());

                result = inputStreamToString1(in);

            } catch (Exception e) {

                e.printStackTrace();

            }

            return result;

        }

        @Override

        protected void onPostExecute(String s) {

            super.onPostExecute(s);

            try {

                if(s!=null) {

//                   Log.e("in method", s);

                    JSONArray jsonObject = new JSONArray(s);

                    tag=jsonObject.getString(1);

                    val=jsonObject.getString(2);

                    String[] vals = val.split(",");

                    String v1 = vals[0];

                    String v2 = vals[1];

                    txtv1.setText(v1);

                    txtv2.setText(v2);

                }else{

                }

            } catch (JSONException e) {

                e.printStackTrace();

            }

        }

    }

    private String inputStreamToString1(InputStream is) {

        String rLine = "";

        StringBuilder answer = new StringBuilder();

        InputStreamReader isr = new InputStreamReader(is);

        BufferedReader rd = new BufferedReader(isr);

        try {

            while ((rLine = rd.readLine()) != null) {

                answer.append(rLine);

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

        return answer.toString();

    }

}