package com.example.secondAndroid;

import android.app.Dialog;

import android.content.Context;

import android.content.Intent;

import android.content.IntentSender;

import android.location.\*;

import android.os.AsyncTask;

import android.os.Bundle;

import android.support.v4.app.FragmentActivity;

import android.support.v4.app.NavUtils;

import android.util.Log;

import android.view.View;

import android.widget.ProgressBar;

import android.widget.Toast;

import com.actionbarsherlock.app.ActionBar;

import com.actionbarsherlock.app.SherlockFragmentActivity;

import com.example.secondAndroid.communication.JSONParser;

import com.example.secondAndroid.location.LocationListenerClass;

import com.google.android.gms.common.GooglePlayServicesNotAvailableException;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.maps.\*;

import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.Marker;

import com.google.android.gms.maps.model.MarkerOptions;

import com.google.android.gms.common.GooglePlayServicesUtil;

import com.google.android.gms.common.ConnectionResult;

import com.google.android.gms.common.GooglePlayServicesClient;

import com.google.android.gms.location.LocationClient;

//import com.google.android.gms.location.LocationListener;

import android.location.LocationListener;

import android.location.Geocoder;

import android.location.LocationManager;

import java.io.IOException;

import java.io.InputStream;

import java.util.List;

import java.util.Locale;

import java.util.Timer;

import android.provider.Settings;

import org.apache.http.HttpEntity;

import org.apache.http.HttpResponse;

import org.apache.http.client.ClientProtocolException;

import org.apache.http.client.methods.HttpGet;

import org.apache.http.impl.client.DefaultHttpClient;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import sun.net.www.http.HttpClient;

import sun.org.mozilla.javascript.internal.json.JsonParser;

/\*\*

\* Created with IntelliJ IDEA.

\* User: chaitanya

\* Date: 2013/08/30

\* Time: 5:23 PM

\* To change this template use File | SettingsActivity | File Templates.

\*/

public class MapActivity extends SherlockFragmentActivity implements GooglePlayServicesClient.ConnectionCallbacks, GooglePlayServicesClient.OnConnectionFailedListener {

private MapView mapView;

private GoogleMap googleMap;

private Bundle bundle;

static final int REQUEST\_GOOGLE\_PLAY\_SERVICES=1001;

double latitude,longitude;

private ProgressBar mActivityIndicator;

LocationManager locationManager;

private final static int CONNECTION\_FAILURE\_RESOLUTION\_REQUEST = 9000;

LocationClient mLocationClient;

private static final int MILLISECONDS\_PER\_SECOND = 1000;

public static final int UPDATE\_INTERVAL\_IN\_SECONDS = 5;

private static final long UPDATE\_INTERVAL = MILLISECONDS\_PER\_SECOND \* UPDATE\_INTERVAL\_IN\_SECONDS;

private static final int FASTEST\_INTERVAL\_IN\_SECONDS = 1;

private static final long FASTEST\_INTERVAL = MILLISECONDS\_PER\_SECOND \* FASTEST\_INTERVAL\_IN\_SECONDS;

LocationRequest mLocationRequest;

boolean mUpdatesRequested;

String markerText="You are here";

LocationListener gpsLocationListener,networkLocationListener;

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.map);

ActionBar actionBar=getSupportActionBar();

actionBar.setDisplayHomeAsUpEnabled(true);

LocationListenerMethod();

if(checkGooglePlayServicesAvailable())

{

try {

mLocationClient=new LocationClient(this,this,this);

mLocationRequest = LocationRequest.create();

mLocationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

mLocationRequest.setInterval(FASTEST\_INTERVAL);

mLocationRequest.setFastestInterval(FASTEST\_INTERVAL);

MapsInitializer.initialize(MapActivity.this);

mapView = (MapView) findViewById(R.id.map);

mapView.onCreate(savedInstanceState);

setUpMapIfNeeded();

mUpdatesRequested = false;

} catch (GooglePlayServicesNotAvailableException e) {

// TODO handle this situation

Toast.makeText(getApplicationContext(),"Maps not found"+e.getMessage(),Toast.LENGTH\_LONG).show();

}

}

else{

Toast.makeText(getApplicationContext(),"Please install latest version of Google Play services on your device",Toast.LENGTH\_LONG).show();

}

}

void LocationListenerMethod()

{

// Define a listener that responds to location updates

gpsLocationListener = new android.location.LocationListener() {

public void onLocationChanged(Location location) {

String msg = "Location updated by GPS. Latitude: " + Double.toString(location.getLatitude()) + "," +" Longitude: "+

Double.toString(location.getLongitude());

if(location.getAccuracy()<=100)

{

//Toast.makeText(getApplicationContext(), "GPS Accuracy: "+String.valueOf(location.getAccuracy()), Toast.LENGTH\_SHORT).show();

changeMarkerPosition(location);

}

}

public void onStatusChanged(String provider, int status, Bundle extras) {

}

public void onProviderEnabled(String provider) {

Toast.makeText(getApplicationContext(), "GPS is now turned on.", Toast.LENGTH\_SHORT).show();

}

public void onProviderDisabled(String provider) {

Toast.makeText(getApplicationContext(), "GPS is now turned off. Mshengu application will not work properly.", Toast.LENGTH\_SHORT).show();

}

};

// Define a listener that responds to location updates

networkLocationListener = new android.location.LocationListener() {

public void onLocationChanged(Location location) {

String msg = "Location updated by Network. Latitude: " + Double.toString(location.getLatitude()) + "," + " Longitude: "+

Double.toString(location.getLongitude());

if(location.getAccuracy()<=100)

{

//Toast.makeText(getApplicationContext(), "Network Accuracy: "+String.valueOf(location.getAccuracy()), Toast.LENGTH\_SHORT).show();

changeMarkerPosition(location);

}

}

public void onStatusChanged(String provider, int status, Bundle extras) {

}

public void onProviderEnabled(String provider) {

Toast.makeText(getApplicationContext(), "Network is now turned on.", Toast.LENGTH\_LONG).show();

}

public void onProviderDisabled(String provider) {

Toast.makeText(getApplicationContext(), "Network is now turned off.", Toast.LENGTH\_LONG).show();

}

};

locationManager = (LocationManager) getApplicationContext().getSystemService(LOCATION\_SERVICE);

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER,2000,10, gpsLocationListener);

locationManager.requestLocationUpdates(LocationManager.NETWORK\_PROVIDER,2000,10, networkLocationListener);

}

public boolean onOptionsItemSelected(com.actionbarsherlock.view.MenuItem item) {

if (item.getTitle().equals("Map"))

{

Intent intent= new Intent(getApplicationContext(),MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(intent);

return true;

}

return super.onOptionsItemSelected(item);

}

private class GetAddressTask extends AsyncTask<Location, Void, String> {

Context mContext;

Location loc;

public GetAddressTask(Context context) {

super();

mContext = context;

}

@Override

protected String doInBackground(Location... params) {

Geocoder geocoder = new Geocoder(mContext, Locale.getDefault());

// Get the current location from the input parameter list

loc = params[0];

if(loc==null)

{

return null;

}

// Create a list to contain the result address

List<Address> addresses = null;

try {

addresses = geocoder.getFromLocation(loc.getLatitude(),

loc.getLongitude(), 1);

} catch (IOException e1) {

Log.e("LocationSampleActivity",

"IO Exception in getFromLocation()");

e1.printStackTrace();

return "no";

//return ("IO Exception trying to get address");

} catch (IllegalArgumentException e2) {

// Error message to post in the log

String errorString = "Illegal arguments " +

Double.toString(loc.getLatitude()) +

" , " +

Double.toString(loc.getLongitude()) +

" passed to address service";

Log.e("LocationSampleActivity", errorString);

e2.printStackTrace();

return "no";

}

// If the reverse geocode returned an address

if (addresses != null && addresses.size() > 0) {

// Get the first address

Address address = addresses.get(0);

String addressText = String.format(

"%s, %s, %s",

// If there's a street address, add it

address.getMaxAddressLineIndex() > 0 ?

address.getAddressLine(0) : "",

// Locality is usually a city

address.getLocality(),

// The country of the address

address.getCountryName());

// Return the text

return addressText;

} else {

return "No address found";

}

}

@Override

protected void onPostExecute(String address) {

// mAddress.setText(address);

if(address==null)

{

}

else {

markerText=address;

if(address.equals("no".toString()))

{

// get JSON data from URL

String add=String.valueOf(latitude)+","+String.valueOf(longitude);

String url="http://maps.google.com/maps/api/geocode/json?address="+add+"&ka&sensor=false";

JSONParser jParser = new JSONParser();

String json = jParser.getJSONFromUrl(url);

//markerText=json;

JSONObject jsonObject;

try {

jsonObject = new JSONObject(json);

markerText = ((JSONArray)jsonObject.get("results")).getJSONObject(0).getString("formatted\_address");

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

googleMap.clear();

mapView.invalidate();

mapView.onResume();

LatLng latLng = new LatLng(loc.getLatitude(),loc.getLongitude());

googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

googleMap.animateCamera(CameraUpdateFactory.zoomTo(15));

googleMap.addMarker(new MarkerOptions().position(new LatLng(loc.getLatitude(), loc.getLongitude())).title(markerText));

}

}

}

@Override

protected void onStart() {

super.onStart();

// Connect the client.

mLocationClient.connect();

}

@Override

protected void onStop() {

try{

locationManager.removeUpdates(gpsLocationListener);

locationManager.removeUpdates(networkLocationListener);

}catch (Exception e)

{

Toast.makeText(this, "Exception: "+e.getMessage(),Toast.LENGTH\_SHORT).show();

}

mLocationClient.disconnect();

super.onStop();

}

@Override

public void onConnected(Bundle dataBundle) {

}

@Override

public void onDisconnected() {

// Display the connection status

Toast.makeText(this, "Disconnected. Please re-connect.",Toast.LENGTH\_SHORT).show();

}

@Override

public void onConnectionFailed(ConnectionResult connectionResult) {

if (connectionResult.hasResolution()) {

try {

connectionResult.startResolutionForResult(this,CONNECTION\_FAILURE\_RESOLUTION\_REQUEST);

} catch (IntentSender.SendIntentException e) {

e.printStackTrace();

}

} else {

showGooglePlayServicesAvailabilityErrorDialog(connectionResult.getErrorCode());

}

}

boolean checkGooglePlayServicesAvailable() {

final int status = GooglePlayServicesUtil.isGooglePlayServicesAvailable(MapActivity.this);

Log.d("google\_play","checkGooglePlayServicesAvailable, connectionStatusCode="

+ status);

if(status != ConnectionResult.SUCCESS)

{

if (GooglePlayServicesUtil.isUserRecoverableError(status)) {

showGooglePlayServicesAvailabilityErrorDialog(status);

return false;

}

}

return true;

}

void showGooglePlayServicesAvailabilityErrorDialog(final int connectionStatusCode) {

MapActivity.this.runOnUiThread(new Runnable() {

public void run() {

final Dialog dialog = GooglePlayServicesUtil.getErrorDialog(connectionStatusCode, MapActivity.this, REQUEST\_GOOGLE\_PLAY\_SERVICES);

if (dialog == null) {

Toast.makeText(getApplicationContext(),

"Incompatible version of Google Play Services",

Toast.LENGTH\_LONG).show();

}

dialog.show();

}

});

}

private void setUpMapIfNeeded() {

if (googleMap == null) {

boolean isGPSEnabled = false,isNetworkEnabled=false;

// getting GPS status

isGPSEnabled = locationManager.isProviderEnabled(LocationManager.GPS\_PROVIDER);

isNetworkEnabled =locationManager.isProviderEnabled(LocationManager.NETWORK\_PROVIDER);

googleMap = ((MapView) findViewById(R.id.map)).getMap();

googleMap.clear();

mapView.invalidate();

mapView.onResume();

googleMap.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);

//Last Known location just to get Map

if(isGPSEnabled)

{

Location location = locationManager.getLastKnownLocation(LocationManager.GPS\_PROVIDER);

if(location!=null)

{

LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());

googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

googleMap.animateCamera(CameraUpdateFactory.zoomTo(14));

Toast.makeText(getApplicationContext(),"Waiting for signal",Toast.LENGTH\_LONG).show();

}

}else if(isNetworkEnabled)

{

Location location = locationManager.getLastKnownLocation(LocationManager.NETWORK\_PROVIDER);

if(location!=null)

{

LatLng latLng = new LatLng(location.getLatitude(),location.getLongitude());

googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

googleMap.animateCamera(CameraUpdateFactory.zoomTo(14));

Toast.makeText(getApplicationContext(),"Waiting for signal",Toast.LENGTH\_SHORT).show();

}

}

else{

Toast.makeText(getApplicationContext(),"Internet is required to load Maps",Toast.LENGTH\_LONG).show();

}

}

}

public void changeMarkerPosition(Location location)

{

if (location != null) {

googleMap.clear();

mapView.invalidate();

mapView.onResume();

latitude = location.getLatitude();

longitude = location.getLongitude();

googleMap.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);

LatLng latLng = new LatLng(latitude, longitude);

googleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

googleMap.animateCamera(CameraUpdateFactory.zoomTo(15));

googleMap.addMarker(new MarkerOptions().position(new LatLng(latitude, longitude)).title(markerText));

(new GetAddressTask(getApplicationContext())).execute(location);

}

}

}