**CS560 - Knowledge Discovery and Management**

**Lab2 – Assignment**

**Problem Statement:**

Complete the following tasks:

1. Implement a Mobile Client Application using either

1.  Mobile Web App Technologies (Javascript, HTML5, CSS3, WebGL, jQuery Mobile (Refer to Tutorial 2))

1. OR Android App technologies (refer to Tutorial 3)

Also refer to Google Chrome Experiences Site for 3D Apps

2. Also use existing Web Services (e.g., Google Map Services, Weather Services) for the application.

3. Write a short report on your work (including screenshots).

4. Post all your work (source code and report) to Lab 2 directory of your GitHub site. And share your GitHub Lab 2 link below.

**Procedure:**

Developed an application in Android App technology.

**Significance:**

While driving or to tell someone the address of the current location is often difficult. This application will tell the current address of the location. Additionally this application gives the current date and time, weather conditions with images on 1 click.

**Use of the Application:**

**Automatically gets the current address, climate without entering any details.**

1. Finds the current latitude and longitude with the GPS locator automatically.
2. Gives the current address location by using the google api , Eg : <http://maps.googleapis.com/maps/api/geocode/json?latlng=39.037176,-94.6666&sensor=true>
3. Gives the current weather condition of the location using the google api,Eg: <http://api.wunderground.com/api/36b799dc821d5836/conditions/q/%22KS/Mission.json>
4. Animation: Rolling an image from left to right and right to left.
5. Gives the Current date and Time.

**Learnt:**

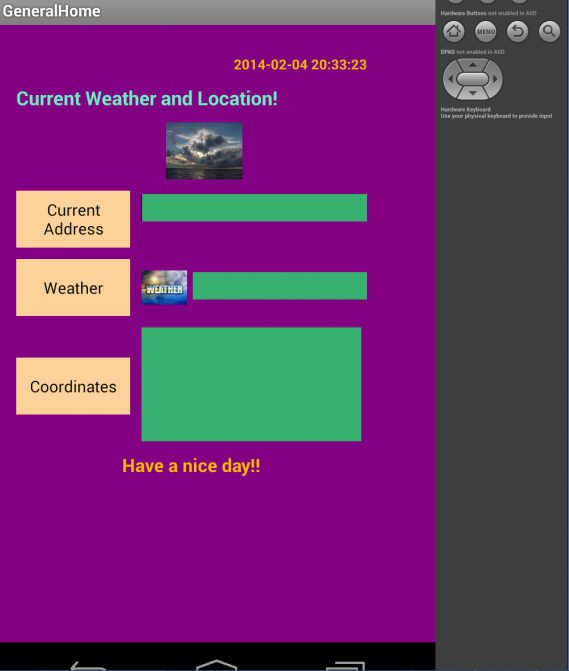
Working with JSON, retrieving JSON data in Android.

HashMap to save the weather details.

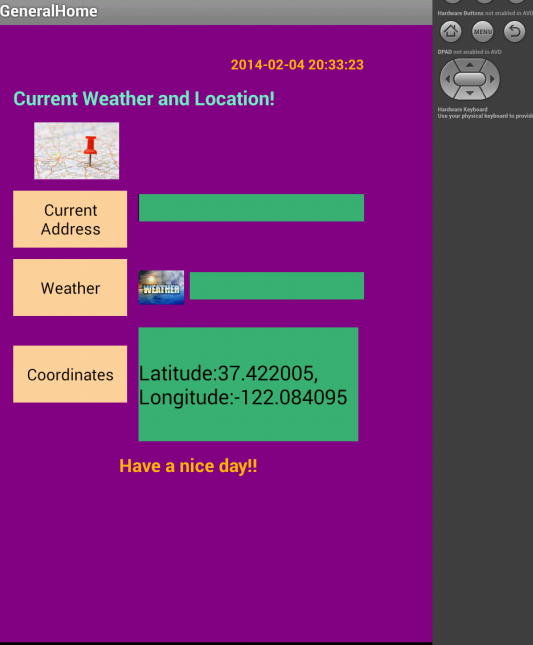
Animation in Android (rolling the image in axis).

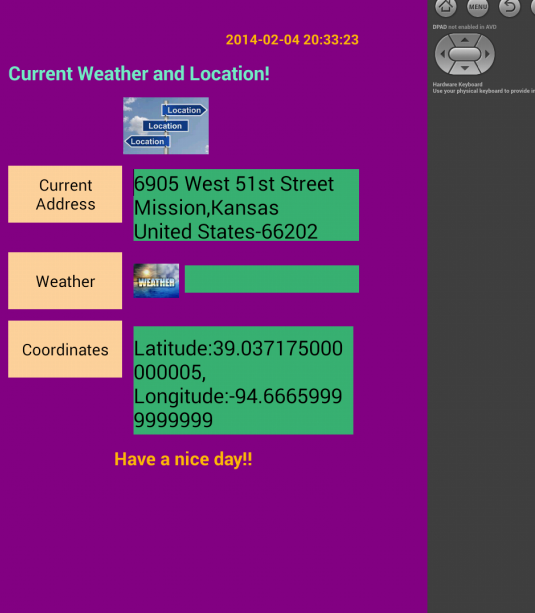
**Screenshots:**

**App Page:**



**Latitude and longitude:**



**Current Address:**

**Current Weather:**



**To Run on Android Emulator:**

**Give the latitude and longitude manually:**

**Windows->Open perspective -> DDMS**

**In Location Controls-> Manual-> Enter the latitude and longitude ->Send**

**Source code: (Project on github also)**

**Activity.xml:**

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

android:paddingBottom=*"@dimen/activity\_vertical\_margin"*

android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*

android:paddingRight=*"@dimen/activity\_horizontal\_margin"*

android:paddingTop=*"@dimen/activity\_vertical\_margin"*

tools:context=*".MainActivity"* >

<TableLayout

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginRight=*"44dp"* >

<TableRow

android:id=*"@+id/tableRow0a"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"10dp"* >

<TextView

android:id=*"@+id/date"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"right"*

android:text=*"date"*

android:textColor=*"#ffb90f"*

android:textSize=*"12dp"*

android:textStyle=*"bold"* />

</TableRow>

<TableRow

android:id=*"@+id/tableRow0"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"10dp"* >

<TextView

android:id=*"@+id/textView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_gravity=*"left"*

android:text=*"@string/heading"*

android:textSize=*"17dp"*

android:textColor=*"#76eec6"*

android:textStyle=*"bold"* />

</TableRow>

<TableRow

android:id=*"@+id/tableRow1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"200dp"*

android:layout\_marginTop=*"10dp"* >

<ImageView

android:id=*"@+id/page\_images"*

android:layout\_width=*"100dp"*

android:layout\_height=*"50dp"*

android:src=*"@drawable/top1"* />

</TableRow>

<TableRow

android:id=*"@+id/tableRow2a"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"10dp"* >

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/add"*

android:layout\_width=*"100dp"*

android:layout\_height=*"50dp"*

android:layout\_marginRight=*"10dp"*

android:background=*"#ffd39b"*

android:text=*"Current Address"* />

<EditText

android:id=*"@+id/address"*

android:layout\_width=*"200dp"*

android:layout\_height=*"wrap\_content"*

android:layout\_weight=*"1"*

android:background=*"#3cb371"*

android:ems=*"10"* />

</LinearLayout>

</TableRow>

<TableRow

android:id=*"@+id/tableRow3"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"10dp"* >

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/weather"*

android:layout\_width=*"100dp"*

android:layout\_height=*"50dp"*

android:layout\_marginRight=*"10dp"*

android:background=*"#ffd39b"*

android:text=*"Weather"* />

<ImageView

android:id=*"@+id/weather\_img"*

android:layout\_width=*"40dp"*

android:layout\_height=*"50dp"*

android:layout\_marginRight=*"5dp"*

android:src=*"@drawable/weather"* />

<EditText

android:id=*"@+id/weather\_text"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:background=*"#3cb371"*

android:ems=*"10"* />

</LinearLayout>

</TableRow>

<TableRow

android:id=*"@+id/tableRow4"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"10dp"* >

<LinearLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:orientation=*"horizontal"* >

<Button

android:id=*"@+id/latlng"*

android:layout\_width=*"100dp"*

android:layout\_height=*"50dp"*

android:layout\_marginRight=*"10dp"*

android:background=*"#ffd39b"*

android:text=*"Coordinates"* />

<EditText

android:id=*"@+id/latLng\_text"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"100dp"*

android:layout\_marginEnd=*"5dp"*

android:background=*"#3cb371"*

android:ems=*"10"* />

</LinearLayout>

</TableRow>

<TableRow

android:id=*"@+id/tableRow0a"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_marginTop=*"10dp"* >

<RelativeLayout

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"* >

<TextView

android:id=*"@+id/day"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_centerInParent=*"true"*

android:text=*"Have a nice day!!"*

android:textColor=*"#ffb90f"*

android:textSize=*"16dp"*

android:textStyle=*"bold"* />

</RelativeLayout>

</TableRow>

</TableLayout>

</RelativeLayout>

**MainActivity.java**

package com.example.generalhome;

import java.io.ByteArrayOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.net.URL;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.HashMap;

import java.util.List;

import java.util.Locale;

import org.apache.http.HttpResponse;

import org.apache.http.HttpStatus;

import org.apache.http.StatusLine;

import org.apache.http.client.ClientProtocolException;

import org.apache.http.client.HttpClient;

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

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

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import android.location.Address;

import android.location.Criteria;

import android.location.Geocoder;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Build;

import android.os.Bundle;

import android.os.StrictMode;

import android.annotation.SuppressLint;

import android.annotation.TargetApi;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.graphics.drawable.Drawable;

import android.text.TextUtils;

import android.util.Log;

import android.view.Menu;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.Animation.AnimationListener;

import android.view.animation.TranslateAnimation;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.TextView;

@TargetApi(Build.VERSION\_CODES.GINGERBREAD)

@SuppressLint("NewApi")

public class MainActivity extends Activity implements LocationListener{

@SuppressLint("NewApi")

StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();

private Button weatherButton;

ImageView image ;

private EditText weatherText;

private EditText latLngText;

private Button latLngButton;

private Button AddressButton;

private String Address1 = "", Address2 = "", City = "", State = "", Country = "", County = "", PIN = "";

private EditText addressText;

private TextView date;

private double latitude;

private double longitude;

protected LocationManager locationManager;

protected LocationListener locationListener;

protected Context context;

private String date\_text;

private ImageView weather\_img;

private String lat,lng;

private int[] imageids;

private int count;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

StrictMode.setThreadPolicy(policy);

setContentView(R.layout.activity\_main);

//Initialising variables

AddressButton = (Button) findViewById(R.id.add);

image = (ImageView)findViewById(R.id.page\_images);

latLngButton = (Button)findViewById(R.id.latlng);

weatherButton = (Button) findViewById(R.id.weather);

System.out.println("weater button");

weatherText = (EditText)findViewById(R.id.weather\_text);

System.out.println("latlng");

latLngText = (EditText) findViewById(R.id.latLng\_text);

addressText = (EditText) findViewById(R.id.address);

date\_text = java.text.DateFormat.getDateTimeInstance().format(Calendar.getInstance().getTime());

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

date.setText(date\_text);

weather\_img = (ImageView)(findViewById(R.id.weather\_img));

lat="";

lng="";

count =0;

imageids = new int[]{R.drawable.top9, R.drawable.top3,R.drawable.top7,R.drawable.top5,R.drawable.top10,

R.drawable.top4,R.drawable.top8,R.drawable.top2,R.drawable.top6,R.drawable.top1};

//getting the location with location manager

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

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 0, 0, this);

//Setting Date on right top corner

Calendar c = Calendar.getInstance();

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String formattedDate = df.format(c.getTime());

date.setText(formattedDate) ;

//Animation for the image

TranslateAnimation animation = new TranslateAnimation(-200.0f, 200.0f, 0.0f, 0.0f);

animation.setDuration(5000);

animation.setRepeatCount(Animation.INFINITE);

animation.setRepeatMode(2);

animation.setFillAfter(true);

animation.setAnimationListener(new AnimationListener() {

public void onAnimationStart(Animation animation) {

// TODO Auto-generated method stub

}

public void onAnimationRepeat(Animation animation) {

if(count == imageids.length)

count =0;

image.setImageResource(imageids[count++]); // images changes

}

public void onAnimationEnd(Animation animation) {

}

});

image.startAnimation(animation);

//Address button gives the current address

AddressButton.setOnClickListener(new View.OnClickListener(){

public void onClick(View v){

getlatlng();

getAddress();

String fullAddress = ""+Address1+Address2+"\n"+City +","+State+"\n"+Country+"-"+PIN;

addressText.setText(fullAddress);

}});

//gives the current weather

weatherButton.setOnClickListener(new View.OnClickListener(){

public void onClick(View v){

getlatlng();

getAddress();

String uri = "http://api.wunderground.com/api/36b799dc821d5836/conditions/q/" +State +"/"+City +".json";

String respString = getJsonFromUrl(uri);

String weather ="";

HashMap<String, String> results = weatherFromJsonResult(respString);

for (HashMap.Entry<String,String> entry : results.entrySet()) {

if (entry.getKey()== "icon\_url"){

try {

InputStream is = (InputStream) new URL(entry.getValue()).getContent();

Drawable d = Drawable.createFromStream(is, "src name");

weather\_img.setImageDrawable(d);

} catch (Exception e) {

}

}

else{

weather = weather + entry.getKey()+" : "+entry.getValue()+"\n";

}

}

weatherText.setText(weather);

}

});

}

@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;

}

//Retriving weather data

public HashMap<String, String> weatherFromJsonResult(String jsonString)

{

HashMap result= new HashMap<String, String>();

String weather ="";

String temp\_f="";

String temp\_c="";

String feels = "";

String icon\_url="";

JSONObject jsonObj1;

try {

jsonObj1 = new JSONObject(jsonString);

JSONObject resultClimate=jsonObj1.getJSONObject("current\_observation");

result.put("Weather",resultClimate.getString("weather"));

result.put("Temp in F",resultClimate.getString("temp\_f"));

result.put("Temp in C",resultClimate.getString("temp\_c"));

result.put("Feels Like",resultClimate.getString("feelslike\_string"));

result.put("icon\_url",resultClimate.getString("icon\_url"));

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return result;

}

//Location manager implemented methods

@Override

public void onLocationChanged(Location location) {

latitude = location.getLatitude();

longitude = location.getLongitude();

System.out.println("location changed");

latLngText.setText("Latitude:" + latitude + ", Longitude:" + longitude);

}

@Override

public void onProviderDisabled(String provider) {

Log.d("Latitude","disable");

}

@Override

public void onProviderEnabled(String provider) {

Log.d("Latitude","enable");

}

@Override

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

Log.d("Latitude","status");

}

//getting JSON from URL

String getJsonFromUrl(String url)

{

HttpClient httpclient = new DefaultHttpClient();

HttpResponse response;

String responseString = null;

try {

HttpPost httpPost = new HttpPost(url);

response = httpclient.execute(httpPost);

StatusLine statusLine = response.getStatusLine();

if(statusLine.getStatusCode() == HttpStatus.SC\_OK){

ByteArrayOutputStream out = new ByteArrayOutputStream();

response.getEntity().writeTo(out);

out.close();

responseString = out.toString();

} else{

//Closes the connection.

response.getEntity().getContent().close();

throw new IOException(statusLine.getReasonPhrase());

}

} catch (ClientProtocolException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}catch (Exception e) {

e.printStackTrace();

}

return responseString;

}

void getlatlng()

{

String latlng[] = latLngText.getText().toString().split(",");

lat =latlng[0].split(":")[1];

lng =latlng[1].split(":")[1];

}

void getAddress()

{

String uri = "http://maps.googleapis.com/maps/api/geocode/json?latlng=" + lat + ","+ lng + "&sensor=true";

String responseString = getJsonFromUrl(uri);

AddressFromJsonResult(responseString);

}

void AddressFromJsonResult(String responseString)

{

try{

JSONObject jsonObj = new JSONObject (responseString);

String Status = jsonObj.getString("status");

if (Status.equalsIgnoreCase("OK")) {

JSONArray Results = jsonObj.getJSONArray("results");

JSONObject zero = Results.getJSONObject(0);

JSONArray address\_components = zero.getJSONArray("address\_components");

for (int i = 0; i < address\_components.length(); i++) {

JSONObject addresses = address\_components.getJSONObject(i);

String long\_name = addresses.getString("long\_name");

JSONArray mtypes = addresses.getJSONArray("types");

String Type = mtypes.getString(0);

if (TextUtils.isEmpty(long\_name) == false || !long\_name.equals(null) || long\_name.length() > 0 || long\_name != "") {

if (Type.equalsIgnoreCase("street\_number")) {

Address1 = long\_name + " ";

} else if (Type.equalsIgnoreCase("route")) {

Address1 = Address1 + long\_name;

} else if (Type.equalsIgnoreCase("sublocality")) {

Address2 = long\_name;

} else if (Type.equalsIgnoreCase("locality")) {

// Address2 = Address2 + long\_name + ", ";

City = long\_name;

System.out.println("cityyyyy"+City);

} else if (Type.equalsIgnoreCase("administrative\_area\_level\_2")) {

County = long\_name;

} else if (Type.equalsIgnoreCase("administrative\_area\_level\_1")) {

State = long\_name;

System.out.println("state" +State);

} else if (Type.equalsIgnoreCase("country")) {

Country = long\_name;

} else if (Type.equalsIgnoreCase("postal\_code")) {

PIN = long\_name;

}

}

}

}

} catch (JSONException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}