

ภาคผนวก

ภาคผนวก ก

**User interface**

หน้าแรกของโปรแกรม



กำหนดพื้นที่



แสดงหน้าสำรวจสัญญาณไวไฟ



แสดงหน้าการวิเคราะห์ wireless



กำหนดขอบเขตค่าความละเอียดช่อง



แสดงผลการวิเคราะห์



แสดงหน้าการตั้งค่าการใช้งาน



แสดงหน้าวิธีการใช้งาน



ภาคผนวก ข

โค้ด โปรแกรม

```
package com.example.wirelessurveies;

import java.util.List;

import android.app.Activity;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.IntentFilter;

import android.content.DialogInterface.OnClickListener;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.net.wifi.ScanResult;

import android.net.wifi.WifiInfo;

import android.net.wifi.WifiManager;

import android.os.Bundle;

import android.provider.Settings;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;
```

```
import android.widget.CompoundButton;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class Survey extends Activity implements OnClickListener,

        android.view.View.OnClickListener {

    locationManager mL;

    LocationListener myLocationListener;

    double latt;

    double lonn;

    boolean dArray = false;

    public static double rad = 0.000008984, d, res;

    // rad=0.000008984;

    public static double maxlat = 0, minlat = 0, maxlon = 0, minlon = 0;

    TextView lat, lon, tv1, tv4, tv5;

    double i;

    double[] latitude = new double[4];

    double[] longitude = new double[4];

    private Button buttonMark;

    private EditText EditText1;
```

```
@Override

public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    // gps-----

    mL = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

    myLocationListener = new MyLocationListener();

    setContentView(R.layout.activity_survey);

    mL.requestLocationUpdates(LocationManager.GPS_PROVIDER, 500, 0,

        myLocationListener);

    // -----

    tv1 = (TextView) findViewById(R.id.textView1);

    tv4 = (TextView) findViewById(R.id.textView4);

    tv5 = (TextView) findViewById(R.id.textView5);

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

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

    lat.setText("");

    lon.setText("");

    EditText1 = (EditText) findViewById(R.id.editText1);
```

```

        buttonMark = (Button) findViewById(R.id.button_mark);

        buttonMark.setOnClickListener(this);

    }

    public boolean isGpsEnable() { // เมฆอดตรวจสอบว่าเปิด GPS หรือไม่

        boolean isgpsenable = false;

        String provider = Settings.Secure.getString(getContentResolver(),

            Settings.Secure.LOCATION_PROVIDERS_ALLOWED);

        if (!provider.equals("")) { // GPS is Enabled

            isgpsenable = true;

        }

        return isgpsenable;

    }

    // class ใช้งาน GPS

    private class MyLocationListener implements LocationListener {

        public void onLocationChanged(Location loc) {

            double latitude[] = new double[100];

            double longitude[] = new double[100];

```

```
latt = loc.getLatitude();

lonn = loc.getLongitude();

lon.setText(Double.toString(Math.floor(lonn * 1000000) / 1000000));

lat.setText(Double.toString(Math.floor(latt * 1000000) / 1000000));

// lon.setText(Double.toString(Math.floor(lonn* 10000) / 10000));

// lat.setText(Double.toString(Math.floor(latt* 10000) / 10000));

// double inc = 0;

// for (int i = 0; i <= 10; i++) {

// latitude[i] = latt + inc;

// longitude[i] = lonn + inc;

// inc += 0.2;

// }

}

public void onProviderDisabled(String provider) {

}

public void onProviderEnabled(String provider) {

}

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

```

    }
}

```

```

public boolean onCreateOptionsMenu(Menu menu) {

    menu.add(0, 0, 0, "Refresh");

    return super.onCreateOptionsMenu(menu);

}

```

```

public void onClick(View v) {

    i++;

    if (i == 1) {

        latitude[0] = (Math.floor(latt * 1000000) / 1000000);

        longitude[0] = (Math.floor(lonm * 1000000) / 1000000);

        Context context = getApplicationContext();

        CharSequence text = "เพิ่ม latitude longitude จุดที่ 1 เรียบร้อย ไปจุด  
ต่อไปครับ";

        int duration = Toast.LENGTH_SHORT;

        Toast toast = Toast.makeText(context, text, duration);

        toast.show();

```

```

        return;
    }

    else if (i == 2) {

        latitude[1] = (Math.floor(latt * 1000000) / 1000000);

        longitude[1] = (Math.floor(lonn * 1000000) / 1000000);

        Context context = getApplicationContext();

        CharSequence text = "เพิ่ม latitude longitude จุดที่2 เรียบร้อย ไปจุด
ต่อไปครับ";

        int duration = Toast.LENGTH_SHORT;

        Toast toast = Toast.makeText(context, text, duration);

        toast.show();

        return;
    }

    else if (i == 3) {

        latitude[2] = (Math.floor(latt * 1000000) / 1000000);

        longitude[2] = (Math.floor(lonn * 1000000) / 1000000);

```

```

Context context = getApplicationContext();

CharSequence text = "เพิ่ม latitude longitude จุดที่3 เรียบร้อย ไปจุด
ต่อไปครับ";

int duration = Toast.LENGTH_SHORT;

Toast toast = Toast.makeText(context, text, duration);

toast.show();

return;

} else if (i == 4) {

    latitude[3] = (Math.floor(latt * 1000000) / 1000000);

    longitude[3] = (Math.floor(lonn * 1000000) / 1000000);

    Context context = getApplicationContext();

    CharSequence text = "เพิ่ม latitude longitude ครบ4จุดแล้วครับ";

    int duration = Toast.LENGTH_SHORT;

    Toast toast = Toast.makeText(context, text, duration);

    toast.show();

```

```
        return;
    }
}

public void calClick(View v) {
    double xz, yz;
    for (int i = 0; i < latitude.length; i++) {
        if (i == 0) {
            maxlat = latitude[i];
            maxlon = longitude[i];
            minlat = latitude[i];
            minlon = longitude[i];
        }

        if (latitude[i] > maxlat) {
            maxlat = latitude[i];
        }

        if (latitude[i] < minlat) {
            minlat = latitude[i];
        }
    }
}
```

```
    }

    if (longitude[i] > maxlon) {
        maxlon = longitude[i];
    }

    if (longitude[i] < minlon) {
        minlon = longitude[i];
    }

}

res = Double.parseDouble(EditText1.getText().toString());

d = rad * res;

yz = (maxlat - minlat) / d;

if (yz < 0) {
    yz = yz * -1;
}

xz = (maxlon - minlon) / d;

if (xz < 0) {
    xz = xz * -1;
}

tv5.setText(Double.toString(Math.ceil(xz)) + "กิโลเมตร");

tv1.setText(Double.toString(Math.ceil(yz)) + "เมตร");
```

```
Mydata.xx = (Math.ceil(yz)) * (Math.ceil(xz));
```

```
tv4.setText("มีจำนวนทั้งหมด" + Double.toString(Math.ceil(Mydata.xx))  
           + "ช่อง");
```

```
return;
```

```
}
```

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

```
    // TODO Auto-generated method stub
```

```
}; // end Location class 99.8445216 19.9858916 99.8444316 19.9852166
```

```
}
```