



IT CARLOW

Traffic Tracking Application on Android

Xufeng Li

C00131008

Content

1 Background	3
2 Similar applications	3
2.1 Safe Driving [2]	3
2.2 Careful Driver [3]	5
2.3 Speed Camera Locator Trial [4]	7
3 Mapping technologies	10
3.1 How to get maps?	10
3.2 How are maps stored?	14
4. GPS technology	16
5. Technology of development	18
5.1 Android SDK	18
5.2 Java	26
5.3 SQLite	27
5.4 Other Development Technologies in Android	28
6. Reference	30

1 Background

At 16th November 2010, the privately operated mobile speed camera was appointed to patrol the national road network, it as part of Road Safety Authority's focus on reducing road fatalities due to speeding and careless driving. The traffic tracking application provides maps which contain the location of all speed mobile cameras tracking routes [1] in the Ireland. The app will advise drivers when they are traveling within these areas, and warn them to stay within the speed limit. Once driver's speed is over the local speed limit, the app can warn them to slow down. Mover over the data about driving can be transferred to a website, such as the date, time and speed. This can be used to confirm to the drivers whether the speed was exceeded. By using this app, parents can to track their children's driving, or careless drivers to keep save driving in Ireland. In this report, I researched some similar applications and some useful technologies about this project.

2 Similar applications

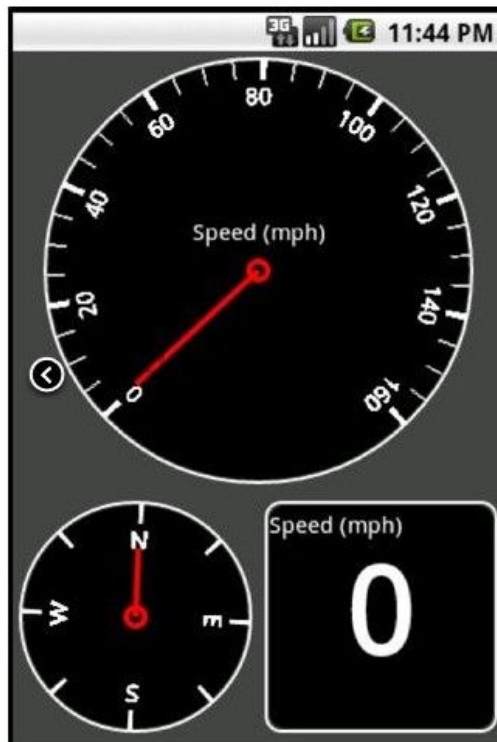
2.1 Safe Driving [2]

Safe Driving is a small application on the Android platform, it allows users to set a speed limit for themselves and alerts when speed bigger than set speed limit. App has some ways to alert drivers, such as vibrate, voice alerts or by flashing the speed dial. Safe Driving can automatically start in background if docked in the car. Here has some screenshots about it.



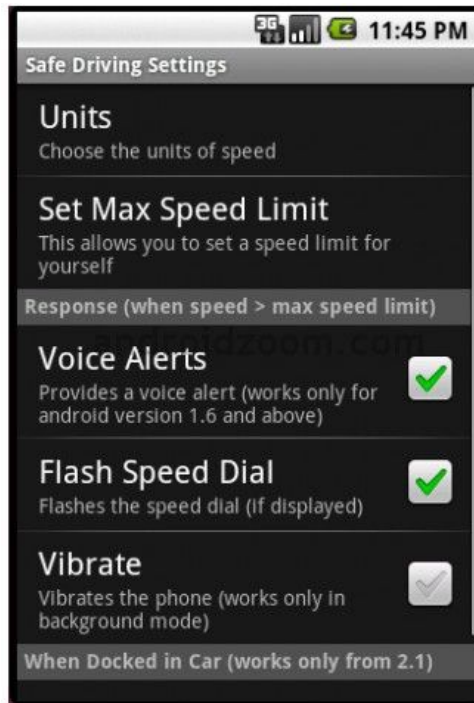
(Safe Driving-1)

In Safe Driving, user can chooses to setting, start monitoring or start monitoring in background.



(Safe Driving-2)

In Safe Driving-2, system displays speedometer and compass.

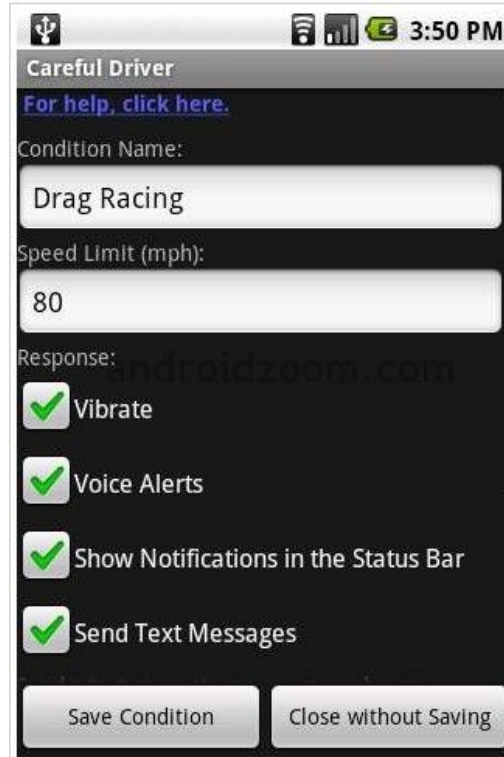


(Safe Driving-3)

In Safe Driving-3, it shows the setting menu which about choose the units of speed, set the way of alerts, etc.

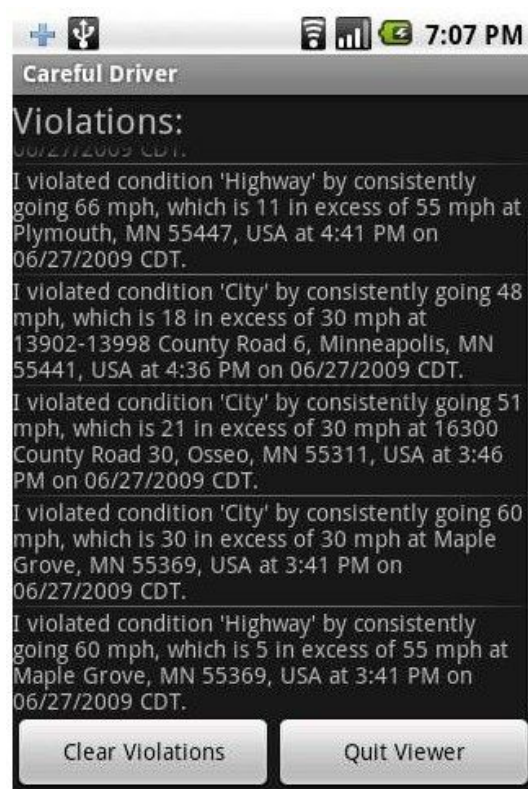
2.2 Careful Driver [3]

Careful Driver is a powerful application designed to help driver keep safely drive. It helps drivers stay under a speed which they specify. The application uses GPS to determine speed and measures a sustained speed for a reasonable period of time. If the measured speed is more than user-defined speed limit, it will notify drivers in a variety of ways, such as phone vibration, voice alerts. All of violations can send text messages to other phones. This is screenshots about it.



(Careful Driver-1)

It shows some setting in Careful Driver, name, speed limit and way of notify.



(Careful Driver-2)

In Careful Driver-2, it displays some of violations about this driver. Any one of it can send to other phones.

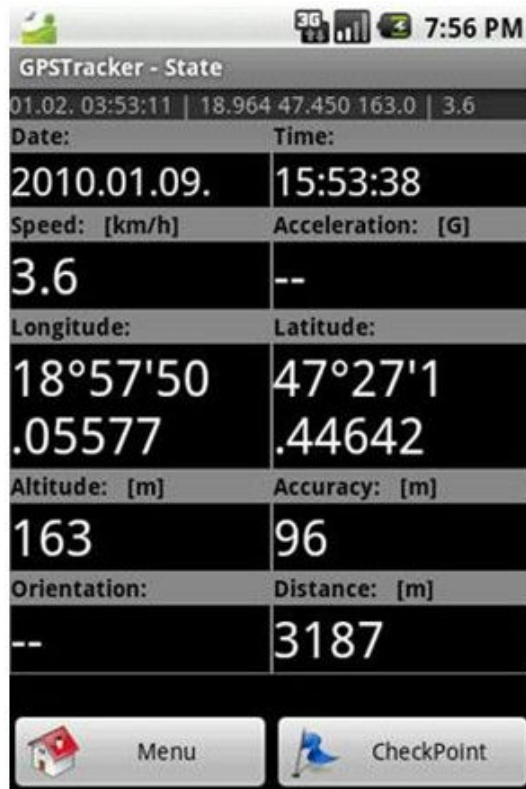
2.3 GPS Tracker Lite [4]

The GPS Tracker Lite is a lightweight, easy to use GPS route tracker, recorder and analyzer. The program provides extensive information on the current situation, e.g. speeds, cords, etc. and provides graphical statistics, such as speed changes, altitude changes, map, etc.



(GPS Track-1)

This main menu of app, user can create a new GPS log, add a checkpoint, setting, manage logs, etc.



(GPS Tracker-2)

After user click 'State' button, app shows some data which includes GPS coordinates, altitude, speed, GPS accuracy, traveled distance and so on.



(GPS Track-3)

User can view the speed on a time axis, including minimum, maximum and average.



(GPS Track-4)

All of GPS data can be added to Google Maps as an overlay.

3 Mapping technologies

3.1 How to get maps?

There are some ways to get free maps for project.

- ✧ Use Google map API [5]
- ✧ Draw map by myself [6]
- ✧ Find the free offline map package

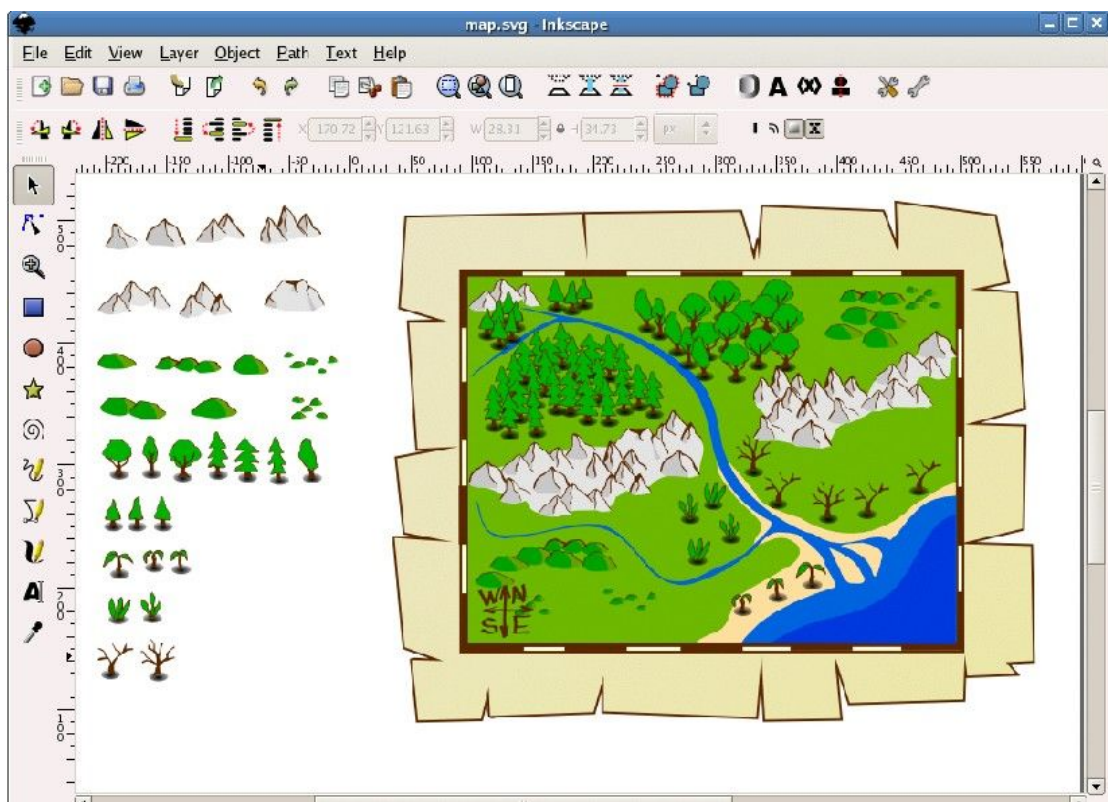
Google map API provides a maps external library for users to add powerful mapping capabilities to their applications. The classes of maps library offer built-in downloading, caching of maps tiles, displays options and controls. It allows applications to manipulate Google Maps data through Map class methods. Here is an example about Google map API in the android platform.



(Google Map API)

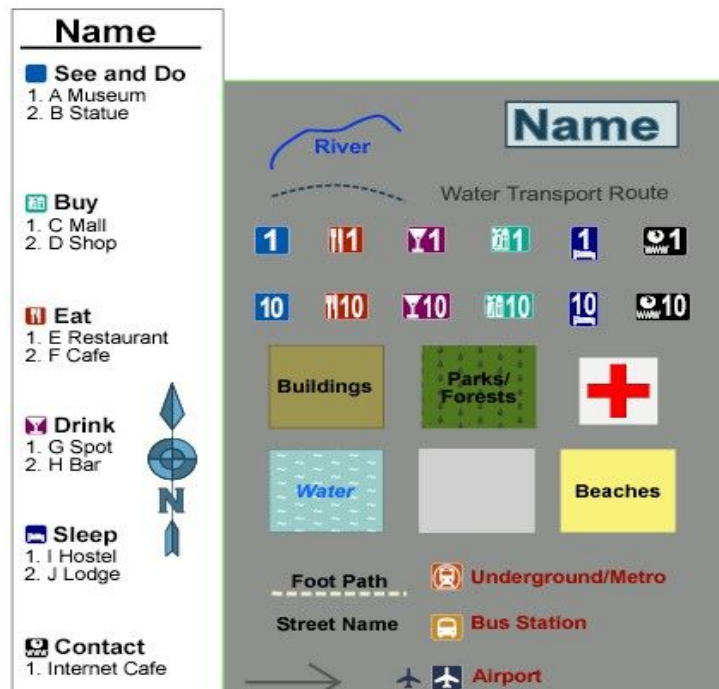
The Google map API is free to use, map servers can quick calculate a best route for applications, and however it requires to full internet access for connecting with map servers to download maps and get traffic data. In other words, if cell phone without internet, their applications could not work for users. So this is not very suitable when users in some area without Internet or some countries where mobile internet might be expensive or not available at all.

I also can choose to draw maps by myself, such as I can draw a small map about IT Carlow, application can use it to achieve some functions of requested. I need some tools which are vector-based drawing tools to build maps. The best one of them is Lnkscope. [7] It is an open-source (free) tool, for Windows, Mac OS X, and Linux. Lnkscope is very good at draw vector graphics. Lnkscope_1 is a screenshot about draws game map.



(Lnkscope_1)

Before begin to draw an actual map, there needs to have templates for map. For example a detailed map of city is created by using the small city templates.

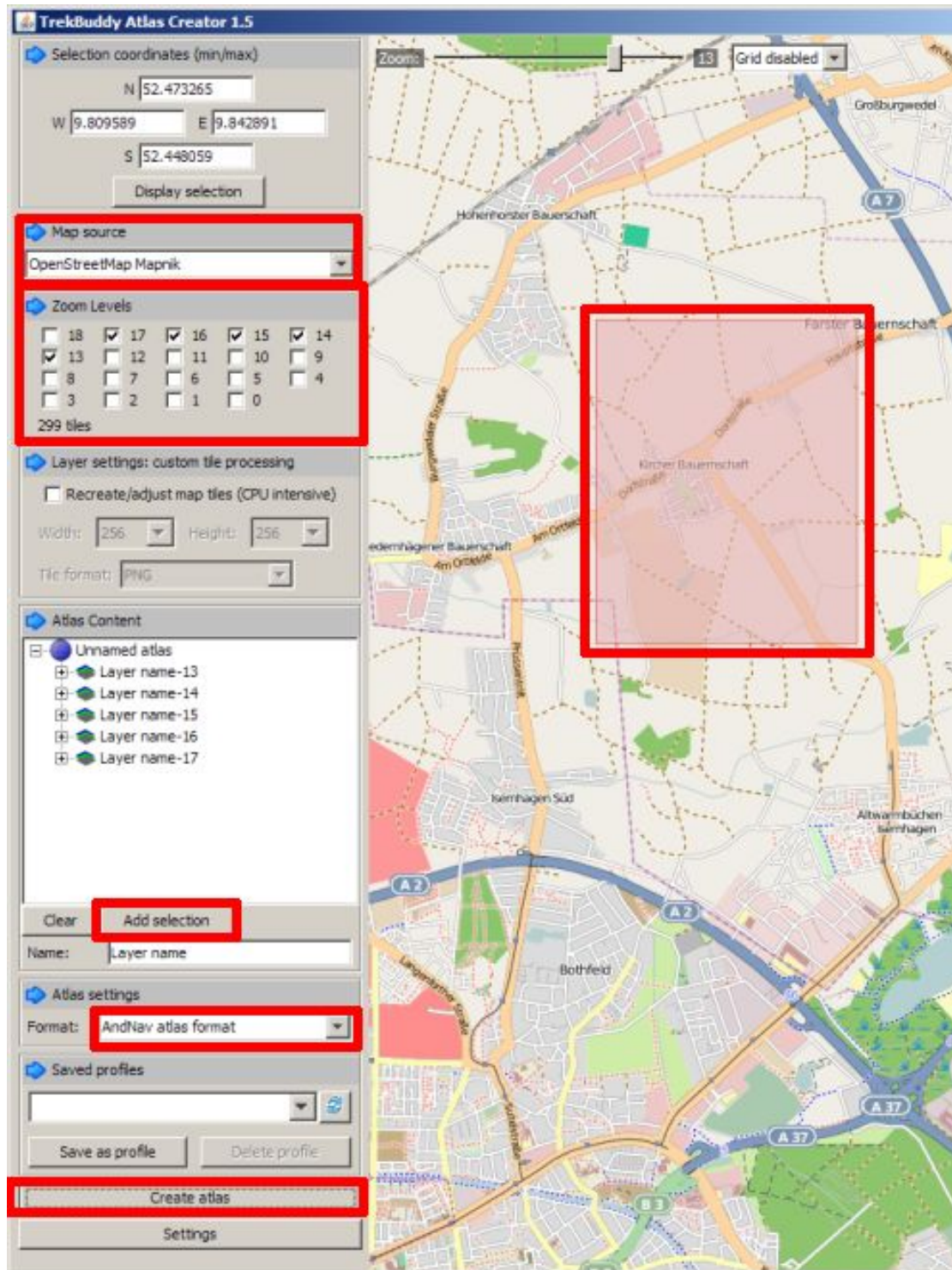


(The small city templates)

These templates can be used in Inkscape for drawing. After getting templates, a base image for the map is also very important. It can be an existing SVG or base image. This image should be imported into a new layer as 'layer Base' which can be used for creating the map on it. After that, there are still some steps to create the map: define the background, define the main area, build the street grid, draw the geography, add buildings, labeling, etc. This kind of map is free to use, and the application does not need to access the internet. However, the function of the project could be limited by the map.

The third way is to find a free offline map package from the internet. There is a tool named 'Mobile Atlas Creator' [8] which is an open source program. It can create offline atlases for GPS handhelds and cell phones. Mobile Atlas Creator can use many different online maps to build offline atlases, such as OpenStreetMap, Google Maps, Microsoft/Bing Maps, and Yahoo Maps. There

is an example screenshot in Mobile Atlas Creator_1.



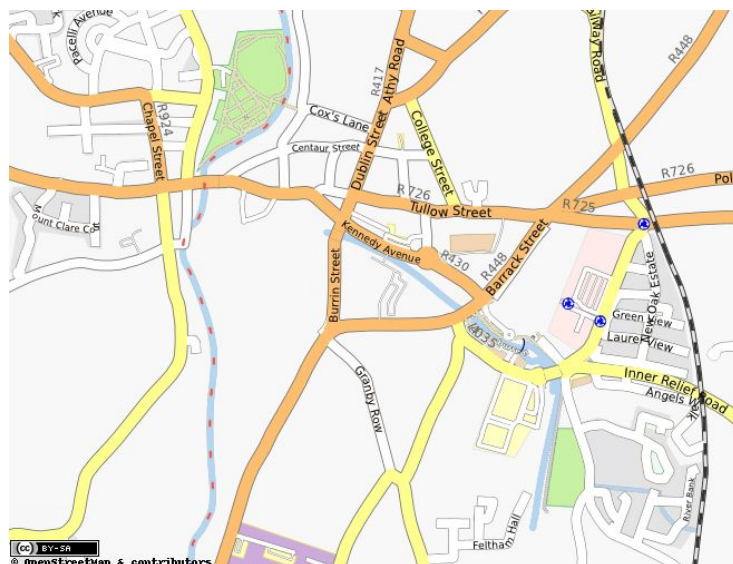
(Mobile Atlas Creator_1)

Users choose the map sources (e.g. OpenStreetMap, Google, etc), the area of map which want to download, the zoom leaves and format of atlas, then click 'create atlas' button to create offline map package. This kind of offline map is easily for use and cell phone does not need to access internet for downloading

maps. This is very useful if users in arrears without network or in foreign countries where mobile internet might be expensive or not available at all. However this is also can limit the function of project and the size of offline map package increases with map's range.

3.2 How are maps stored?

In the Dynamically Updating SAT-NAV system, software needs to store some data about map. For example, when user updates a new street, software stores this new street into database and shows it on the map. Software also can save Carlow's map and use it for calculating the route. That is impossible for normal database to store map or a street. However the spatial database can do it. The spatial database is a database that is optimized to store and query data related to objects in space, including points, lines and polygons. [9] All of these kinds spatial data can be found in the map, here is an example of Google map:



(Google map)

In this map, the spatial data are obviously shown on it. The spatial database not only can be used to stored data, but also has a variety of spatial operations,

such as finds the distance or calculates the area between points. So it is widely used in mapping technology. One of the most famous database systems is PostGIS. PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relation database. [10] Data from a PostGIS/PostgreSQL database can be used as data source for spatial server software like MapServer and GeoServer. Developer use SQL language to manipulate it, for example, these codes show how to insert coordinates by using SQL in PostGIS.

```
Create table user_locations (gid int4, user_name varchar); //create table
Select AddGeometryColumn ('db_mapbender', 'user_locations', 'the_geom', '4326',
'point'); //add column for table
Insert into user_locations vaules ('1','Carlow', GeometryFromText ('POINT
(-71.060316 48.432044)' 4326)); //insert coordinates into table
```

Here also has some other's spatial data type which PostGIS supports:

MULTIPOINT(10 20, 30 40, 50 60)

Specifies a composite element that contains one or more points

LINestring(10 20, 12 21, 13 31)

Specifies a single line

MULTILINestring((10 20, 12 21, 13 31),(20 30, 32 41, 53 61))

Specifies a composite element that contains one or more lines

POLYGON((10 20, 40 50, 40 70, 10 20))

Specifies a single polygon

MULTIPOLYGON(((10 20, 40 50, 40 70, 10 20)),((20 30, 32 41, 53 61)))

Specifies a composite element that contains one or more polygons

By using spatial database, there is a widely field of application. Such as web mapping applications and satellite imagery are strongly increasing in this technology. So the Dynamically Updating SAT-NAV system also can choose it as mapping technology.

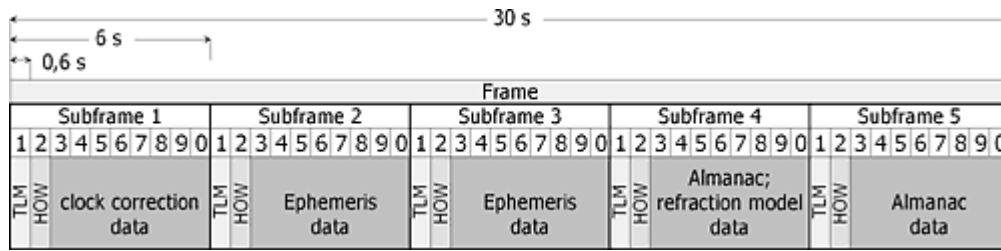
4. GPS technology

“The Global Positioning System (GPS) is a space-based global navigation satellite system that provides reliable location and time information in all weather and at all times and anywhere on or near the Earth when and where there is an unobstructed line of sight to four or more GPS satellites. It is maintained by the United States government and is freely accessible by anyone with a GPS receiver... GPS was created and realized by the U.S Department of Defense and was originally run with 24 satellites. It was established in 1973 to overcome the limitations of previous navigation system.

“[11] The first satellite navigation system was used by the United States Navy in 1960. It used a constellation of five satellites and could provide a navigational fix approximately once per hour. The GPS consists of three parts: the space segment, the control segment and the user segment. The U.S Air Force develops, maintains and operates the space and control segments. GPS satellites broadcast signal from space, which each GPS receiver uses to calculate its three-dimensional location (latitude, longitude and altitude) plus the current time. During GPS receiver is moving, the GPS application can get details of the location and store these points with its time. After that the resulting track can be displayed as a series of the recorded points or a line connecting them.

The navigational signals transmitted by GPS satellites encode a variety of information including satellite positions, the state of the internal clocks and the health of the network. These signals are transmitted on two separate carrier frequencies that are common to all satellites in the network. Each GPS satellite continuously broadcasts a navigation message at a rate of 50 bits per second. Each complete message is composed of 30-second frames, distinct groupings of 1500 bits of information. The data of complete message is divided into 25

frames. There is picture which shows the structure of the GPS data.



(Structure of the GPS data)

The 25 frames are divided into sub frames (300 bit, 6 sec), which are again divided into 10 words each (30 bit, 0.6 sec). The first word of each sub frame is the TLM (telemetry word), which contains the number of counted z-epochs. These data contain the time since last 'restart' of the GPS time on the previous Sunday 0:00 o'clock. The HOW is used by military receivers to locate their access to the P-code. The rest of the first sub frame contains data about status and accuracy of the transmitting satellite as well as clock correction data. The second and third sub frames contain ephemeris parameters. Sub frames 4 and 5 contain the so-called almanac data which include information about orbit parameters of all satellites, their technical status and actual configuration, identification number and so on. [20]

The GPS originated from a military project, so it has significant military and civilian applications. The civilian applications usually used as GPS navigation device which can receive GPS signals for determining the device's current location on Earth, such as hand-held receivers, outdoor receivers, and mobile phone receiver. All of civilian applications contain maps for users and provide suggested directions, information about traffic conditions and information on nearby amenities, such as restaurants, fueling station, etc. The military applications of GPS include some functions, such as help soldiers to find objectives, track ground and air targets, missile and projectile guidance, etc.

5. Technology of development

5.1 Android SDK

The Android SDK (software development kit) includes a variety of custom tools that help users develop mobile applications on the Android platform. [12] These include a debugger, libraries, Android emulator and Android Development Tools which plug-in for Eclipse. It supported development platforms include Linux, Mac OS, Windows XP, Windows Vista or Windows 7. Requirements of SDK also include Java Development Kit, Apache Ant, and Python 2.2 or later. Developers can use any text editor to edit Java and XML files then use command line tools to create, build and debug application. A preview release of the Android SDK was released on 12 November 2007. [13]

There is an example about 'Hello, world' in Android SDK. Firstly the new android project should be started in Eclipse, then create a new java file 'helloandroid.java' and code like this:

```
import android.app.Activity;

import android.os.Bundle;

import android.widget.TextView;

public class HelloAndroid extends Activity {

    /** Called when the activity is first created. */

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
```

```

        TextView tv = new TextView(this);

        tv.setText("Hello, Android");

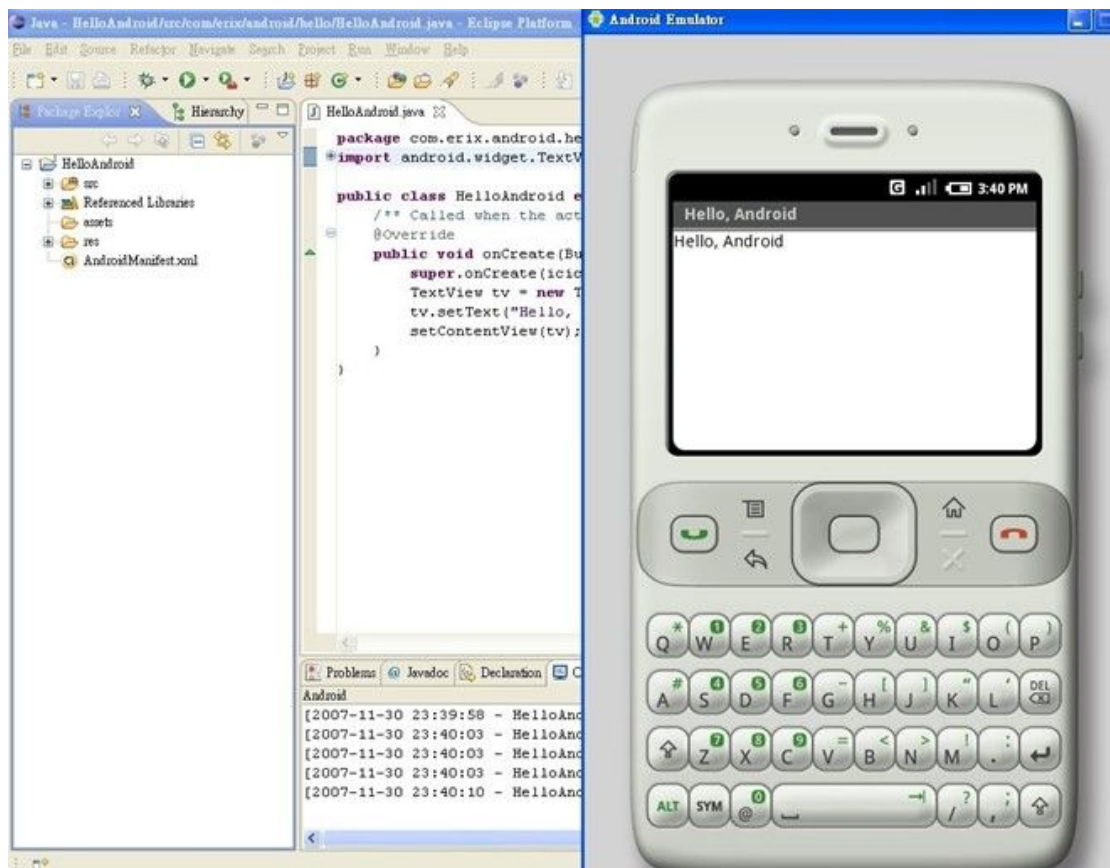
        setContentView(tv);

    }
}

```

Finally user selects Run ->Run -> 'Android Application' to run this application.

We can see the result of it:



(Hello, world)

There is another example about relative layout in Android. Firstly start a new project named 'HelloRelativeLayout', then opens the 'res/layout/main.xml' file and inserts the following codes: [30]

```

<?xml version="1.0" encoding="utf-8"?>

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

    android:layout_width="fill_parent"

    android:layout_height="fill_parent">

```

```

<TextView
    android:id="@+id/label"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Type here:" />

<EditText
    android:id="@+id/entry"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/editbox_background"
    android:layout_below="@id/label" />

<Button
    android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/entry"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10dip"
    android:text="OK" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@id/ok"
    android:layout_alignTop="@id/ok"
    android:text="Cancel" />

</RelativeLayout>

```

After that load this layout in the 'onCreate()' method:

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

```

```
setContentView(R.layout.main);  
}
```

Finally run this application and get result of it:



(HelloRelativeLayout)

The example of 'Hello, world' shows how to use Java code to achieve small application in Android. The second example of 'HelloRelativeLayout' displays that some widgets in Android. Android SDK provides various development packages to users. These packages help developers to make powerful and awesome Android applications. For example the widget package contains UI elements to use on Android's application screen. In 'HelloRelativeLayout', it uses 'TextView', 'EditText' and 'Button' widgets. Here is 'TextView' widgets codes:

```
<TextView  
    android:id="@+id/label"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Type here:" />
```

There are some attributes to describe how developers want to position each

widget. This 'TextView' is defined to lie to the screen with width and height, it also identified by the ID and shown 'Type here'. The widget package has lots of UI elements, such as 'Checkbox', 'CheckedTextView', 'ImageButton', 'ListView' etc.

Developers can build location and maps-based applications by using the classing of the 'android.location' package. [31] The 'android.location' package provides access to the system location services which allow applications to obtain periodic updates of the device's geographical location. There are some classes in the 'android.location' package. The address class can represent an address, the geocoder class for handing geocoding and reverse geocoding, the GpsSatellite class represents the current state of a GPS satellite, the locationManager class provides access to the system location services etc. For example, the following code shows how to define a LocationListener and request location updates:

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {

    public void onLocationChanged(Location location) {

        // Called when a new location is found by the network location provider.
        makeUseOfNewLocation(location);

    }

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

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}

};

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0,
```

```
locationListener);
```

In Location class, it has 'getSpeed' method which is used to get the speed information. The 'hasSpeed' method returns true if speed information is available otherwise false. The speed reported in meters/second.

In the Android, there are two ways to obtain user location. The app can utilize GPS or Android's Network Location Provider to acquire the user location. The GPS is most accurate, but it only works outdoors, quickly consumes battery power and can not return location as quickly as users want. The Android's Network Location Provider gets user location by using cell tower and Wi-Fi signals. It can work indoors and outdoors, responds faster and uses less battery power. At most of time to obtain the user location in the application, developers can use both GPS and the Network Location Provider, or just one. In application, developers can let them share a public listener to achieve GPS and the Network Location Provider work together. Here is a code for use both of them in app.

```
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 10 *  
                                     1000, (float) 10.0, listener);  
  
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 90 *  
1000, (float) 10.0, listener);
```

Menus are an important part of an application that provides a familiar interface for the user to access application function and settings. Android SDK offers a menus interface package for developers to provide application menus in application. [32] The easiest way of adding menu items is inflating an XML file into the 'Menu' via 'MenuInflater'. There are three different types of menu, context menus, options menus and sub menus. For example, here is a code about menu:

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@+id/new_game"

        android:icon="@drawable/ic_new_game"

        android:title="@string/new_game" />

    <item android:id="@+id/quit"

        android:icon="@drawable/ic_quit"

        android:title="@string/quit" />

</menu>

```

This example defines a menu with the following elements:

<menu>

It used to creating a menu which is a container for menu items.

<item>

It used to creating a menu item which represents a single item in a menu.

In some of menus, they use <group> which an optional container for <item> elements.

In this menu also has two menu items, each item includes the attributes:

Android:id

A resource ID that's unique to the item

Android:icon

A draw able resource that is the icon visible to the user

Android:title

A string resource that is the title visible to the user

Android SDK have been a number of updates since its original release. These updates to the base operating system typically fix bugs and add new features. Here is a list of update history of Android. [13]

- **Android 1.5 Platform**

On 30 April 2009, the official 1.5 update for Android was released. It introduces

some new features and changes for users and developers. These changes include user interface refinements, performance improvements, new feature, new APIs and manifest elements. It ability to record and watch videos through camcorder mode, upload videos to You Tube, a new soft-keyboard with text-prediction, Bluetooth A2DP and AVRCP support, etc.

- Android 1.6 Platform

On 15 September 2009, the 1.6 SDK was released. It introduction some new features which about new user features, android market updates and new platform technologies. In the new user features, it supports a quick search box for Android, an integrated camera, camcorder and gallery interface. The Android 1.6 platform brings some new technologies, such as expanded search framework, text to speech engine, gestures, accessibility and Google free turn by turn navigation.

- Android 2.1 Platform

On 12 January 2010, the Android 2.1 was released. It provides some of the new features and technologies which about new user features and new platform technologies. The Android 2.1 optimized hardware speed, supported for more screen size and resolutions, revamped UI, improved Google Maps, and improved virtual keyboard, etc.

- Android 2.2 Platform

On 20 May 2010, the Android 2.2 was released. Its changes included new user features, new platform technologies, new developer services and new developer APIs. It general Android OS speed, memory and performance optimizations, integration of Chrome's V8 JavaScript engine into the Browser application, increased Microsoft Exchange support, added an option to disable data access over mobile network, voice dialing and contact sharing over Bluetooth, support for file upload files in the browser application and Adobe

Flash 10.1 support, etc.

5.2 Java

Java is a programming language developed by James Gosling at Sun Microsystems and released in 1995. This language gets lots of syntax from C and C++ but has a simpler object model and fewer low-level facilities. The Java applications can run on any Java Virtual Machine regardless of computer architecture. Java is one of the most popular programming languages in use, and it widely used from application software to web applications. [14] In this project, I will use Java as development language. There is an example about 'Hello world' in Java:

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```

Java is easy to learn for many of reasons. There have lots of Java resources to help users learn the language, including websites, tutorials and books. Java also is one of the most human-readable languages, which means users can easily to understand Java code. Such as the code in 'Hello world', person can quickly to know what code is doing. After users complete Java codes, the Java compilers convert them from human readable to byte-code which can interpret by a Java Virtual Machine to actually execute the compiled code in different devices environment. The Java Virtual Machine operates code like a physical CPU might operate on machine code. Android applications can run in a special virtual machine called the Dalvik Virtual Machine. [22] Dalvik is an integral part of Android which is typically used on mobile devices such as mobile phones, tablet computers and net books. It can help Android applications to run on any

Dalvik Virtual Machine regardless of the mobile device. So this is the reason why Java applications also can run on the Android platform.

In the Android platform, the Eclipse development environment can compile Java code into byte-code when users save project file. Developer can easily see compile errors during coding. After developer complete Java application in the Android platform, the application package Java files in Android Package which includes compiled Java code and any other resources, such as images, sound files, Android Manifest file. This is can make Java application run well in the Android platform.

5.3 SQLite

SQLite is an ACID-compliant embedded relational database management system contained in a relatively small C programming library. [15] It is the most widely deployed SQL database engine in the world, the source code for SQLite in the public domain and it is free to use for any purpose. [16] SQLite does not have a separate server process. It can read and write directly to ordinary disk files. SQLite is a compact library which size can be less than 300KB and performance is usually quite good even in low-memory environments, so it is widely used in cell phones, PDAs and MP3 players. Every Android application can choose to use SQLite as database. The Android platform supply data API for SQLite, developers can easily use it by SQL languages. Here are some example SQLite codes in the Android platform:

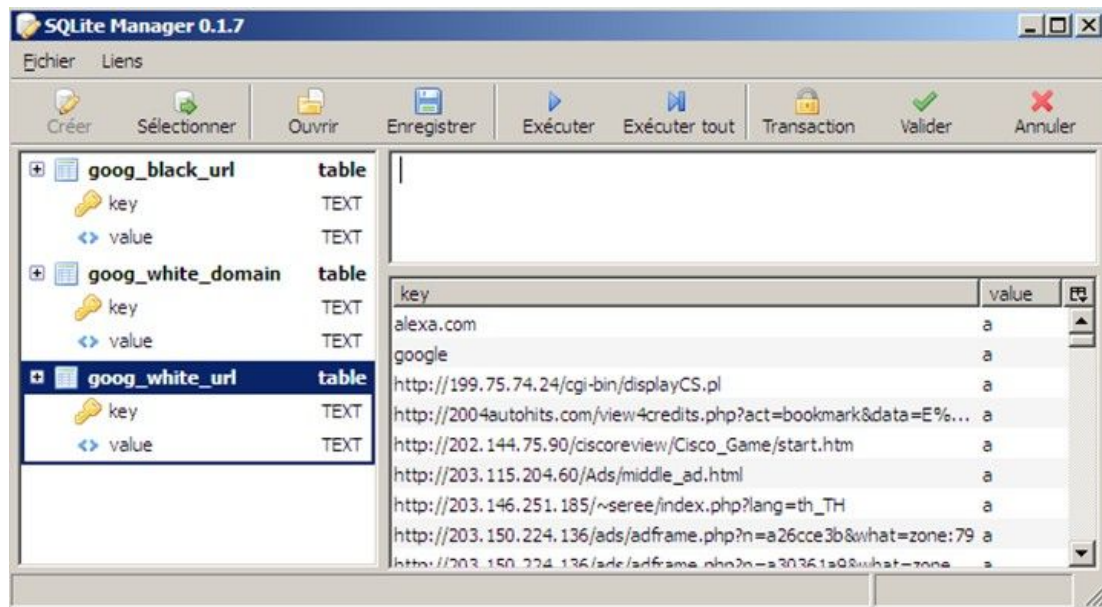
```
db.execSQL("CREATE TABLE mytable (_id INTEGER PRIMARY KEY  
        AUTOINCREMENT, title TEXT, value REAL);");
```

This is use for creating a new table 'mytable'. Its primary key is '_id' which can auto-increment by 1 for each new data.

```
db.execSQL("INSERT INTO widgets (name, inventory)" + "VALUES ('Sprocket', 5)");
```

Application can install new data into table by these codes.

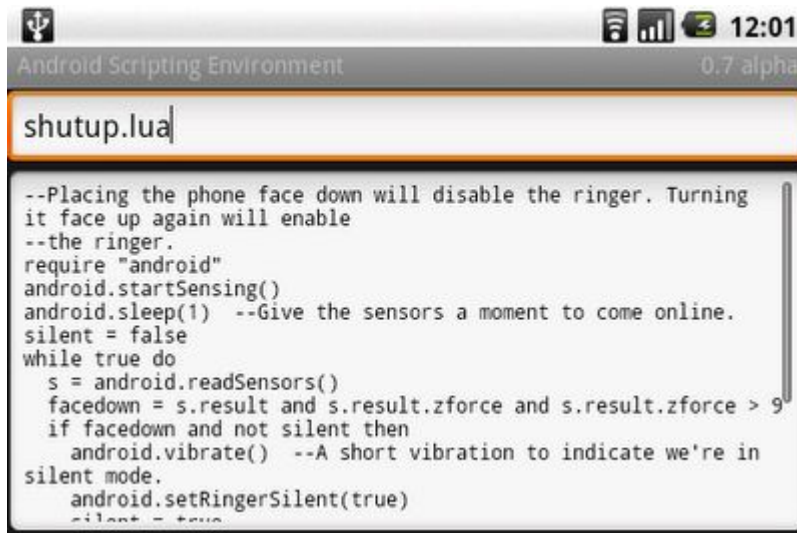
Developers also can use SQLite Manger [17] to manger database. It can manage any SQLite database on computer, supply helpful dialogs to manage table. This is example screenshot of it.



(SQLite Manager)

5.4 Other Development Technologies in Android

Scripting Layer for Android (SL4A) brings scripting languages to Android by allowing you to edit and execute script and interactive interpreters directly on the Android device. These scripts have access to many of the APIs available to full-fledged Android applications, but with a greatly simplified interface that makes it easy to get things done. Scripts can be run interactively in a terminal, started as a long running service, or started via Locale, Python, Perl, JRuby, Lua, BeanShell, JavaScript, Tcl and shell are currently supported. [23]



```
shutup.lua

--Placing the phone face down will disable the ringer. Turning
it face up again will enable
--the ringer.
require "android"
android.startSensing()
android.sleep(1) --Give the sensors a moment to come online.
silent = false
while true do
  s = android.readSensors()
  facedown = s.result and s.result.zforce and s.result.zforce > 9
  if facedown and not silent then
    android.vibrate() --A short vibration to indicate we're in
silent mode.
    android.setRingerSilent(true)
    silent = true
  end
end
```

(Android Scripting-1)

Scripting can be edited directly on the phone. (Android Scripting-1)



(Android Scripting-2)

The script manager displays available scripts. (Android Scripting-2)

SL4A provides high-level support for a subset of Android's APIs which are available to every scripting language (except shell) and are used in the same way. Here are some of examples about the android module for each language.

Python:

```
import android

droid = android.Android()

name = droid.getInput("Hello!", "What is your name?")

print name # name is a namedtuple

droid.makeToast("Hello, %s" % name.result)
```

Lua:

```
Require "android"

name=android.getInput("Hello!", "What is your name?")

android.printDict(name)

android.makeToast("Hello,"..name.result)
```

Perl:

```
Use Android;

My $a = Android->new();

$a->makeToast("Hello, Android!");
```

6. Reference

[1] Map of the mobile camera in Ireland (2010), From an Garda Síochána - Ireland's National Police Service:

<http://garda.ie/gosafe.htm> [accessed 28 November, 2010]

[2] Safe Driving (21 July, 2010), Muthu Lalapet, From Android Zoom:

http://www.androidzoom.com/android_applications/tools/safe-driving_iwrt.html

[accessed 11 November, 2010]

[3] Careful Driver (28 October, 2009), From Android Zoom:
http://www.androidzoom.com/android_applications/tools/careful-driver_uuo.html?nav=related [accessed 11 November, 2010]

[4] GPS Tracker Lite, Gabor Telek (November, 2010), From App Brain:
<http://www.appbrain.com/app/g.android.gpstracker> [accessed 11 November, 2010]

[5] Maps External Library (2010), Google Projects for Android: Google APIs, From Google code:
<http://code.google.com/intl/zh-CN/android/add-ons/google-apis/maps-overview.html>[accessed 25 October, 2010]

[6] Ryan Hollidy, Pert Fitzgerald (9 August, 2010), How to draw a map, From Wikitravel:
http://wikitravel.org/en/Wikitravel:How_to_draw_a_map[accessed 26 October, 2010]

[7] Inkscape (2010), Open Source vector graphics editor, From Inkscape:
<http://inkscape.org/>[accessed 25 October, 2010]

[8] Mobile Atlas Creator (2008), From Mobile Atlas Creator
<http://trekbuddyatlas.sourceforge.net/index.html> [accessed 25 October, 2010]

[9] Spatial database (21 September, 2010), From Wikipedia:
http://en.wikipedia.org/wiki/Spatial_database [accessed 26 October, 2010]

[10] PostGIS (8 October, 2010), From Wikipedia:
<http://en.wikipedia.org/wiki/PostGIS> [accessed 26 October, 2010]

[11] Global Positioning System (2 November, 2010), From Wikipedia:
http://en.wikipedia.org/wiki/Global_Positioning_System [accessed 27 October, 2010]

[12] Tools Overview (2 November, 2010), From Android developers:
<http://developer.android.com/guide/developing/tools/index.html>[accessed 30 October, 2010]

[13] Android (operating system) (3 November, 2010), From Wikipedia:
[http://en.wikipedia.org/wiki/Android_\(operating_system\)#cite_note-86](http://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-86)
[accessed 31 October, 2010]

[14] Java (programming language) (1 October, 2010), From Wikipedia:
[http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language)) [accessed 31
October, 2010]

[15] SQLite (14 October, 2010), From Wikipedia:
<http://en.wikipedia.org/wiki/SQLite>[accessed 1 November, 2010]

[16] About SQLite (2010), From SQLite:
<http://www.sqlite.org/about.html>[accessed 2 November, 2010]

[17] sqlite-manager, Extension for Firefox and other apps to manage any sqlite
database (2010), From Sqlite-manager:
<http://code.google.com/p/sqlite-manager/>[accessed 2 November, 2010]

[18] Dijkstra's algorithm (25 October, 2010), From Wikipedia:
http://en.wikipedia.org/wiki/Dijkstra's_algorithm[accessed 3 November, 2010]

[19] Dijkstra Algorithm Illustrating Greedy (20.07.2007), From Animal Home
Page:
<http://www.animal.ahrgr.de/showAnimationDetails.php3?lang=en&anim=16>[ac
cessed 3 November, 2010]

[20] Composition of the Data Signal (19-04-2009), From Kowoma.da:
<http://www.kowoma.de/en/gps/signals.htm> [accessed 9 November, 2010]

[21] GPS signals (9 October, 2010), From Wikipedia:
http://en.wikipedia.org/wiki/GPS_signals [accessed 9 November, 2010]

[22] Dalvik (2 November, 2010), From Wikipedia:
[http://en.wikipedia.org/wiki/Dalvik_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software)) [accessed 9 November, 2010]

[23] Damon Kohler (8 June, 2009), Introducing Android Scripting Environment,
From Google-open source:
[http://google-opensource.blogspot.com/2009/06/introducing-android-scripting.
html](http://google-opensource.blogspot.com/2009/06/introducing-android-scripting.html) [accessed 10 November, 2010]

[24] Android-scripting (2010), From Google code:
<http://code.google.com/p/android-scripting/> [accessed 10 November, 2010]

[30] Relative Layout (2010), From Android developers:
[http://developer.android.com/resources/tutorials/views/hello-relativelayout.htm
l](http://developer.android.com/resources/tutorials/views/hello-relativelayout.html) [accessed 16 November, 2010]

[31] Obtaining User Location (2010), From Android developers:
<http://developer.android.com/guide/topics/location/obtaining-user-location.html>
[accessed 17 November, 2010]

[32] Creating Menus (2010), From Android developers:
<http://developer.android.com/guide/topics/ui/menus.html> [accessed 17
November, 2010]