DEVELOPING MOBILE APPLICATIONS USING ANDROID

M.VARUN KUMAR, S.THIRUMALINI, S.RAJESH KUMAR, N.VANISRI, PULIDINDI VENUGOPAL

School of Information Technology &VITBS

VIT University,Vellore-632014

memalini467@gmail.com

Abstract-With Google's latest launch for mobile android Operating System, we did a project starting from ABC of it up to Creating Basic application. Here we are describing about Android, features, capabilities, application, what we developed, current market scenario and future plans. With Android, mobile phone has achieved its 3rdgeneration brand in21st century. To cop up with the latest technology in mobile

Index terms-SDK (Software Development Kit),ADT (Android Development Tools),AVD (Android Virtual Device) GPS (Geo Positioning Satellite),API (Application Programming Interface),IDE (Integrated Development Environment) SMS (Short Message service System),lib c – Library

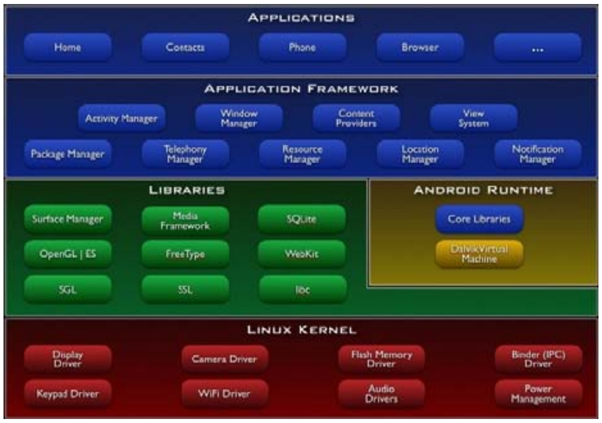
1 INTRODUCTION

Android is a mobile operating system running on the Linux kernel. It was initially developed by Android Inc., a firm later purchased by Google, and lately by the Open Handset Alliance. It allows developers to write managed code in the Java language, controlling the device via Google-developed java libraries.

The unveiling of the Android distribution on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 47hardware, software and telecom companies devoted to advancing open standards for mobile devices. Google released most of the Android code under the Apache License, a free software and open source license.

2 ANDROID ARCHITECTURE

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.



- **System Libraries:** Android includes a set of C/C++libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:
- **System C library:** BSD-derived implementation of the standard C system

library (libc), tuned for embedded Linux-based devices

- **Media Libraries:** Based on Packet Video's Open CORE; the libraries support play back and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR,JPG, and PNG
- **Surface Manager:** Manages access to the display subsystem and seamlessly composites2D and 3D graphic layers from multiple applications
- **Lib Web Core:** A modern web browser engine which powers both the Android browser and an embeddable web view.S GL- the underlying 2D graphics engine.
- **3D libraries:** An implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D
  Acceleration (where available) or the included, highly optimized 3D software rasterizer Free Type- bitmap and vector font rendering
- **SQLite :**A powerful and lightweight relational database engine available to all applications
- **Development Toolkit:**

  •SDK (Software Development Kit)

  •Eclipse

  •ADT Plug in

  •Draw Droid

2.1 Android SDK:

The Android SDK includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator (based on QEMU),documentation, sample code, and tutorials. Currently supported development platforms include x86- architecture computers running Linux (any modern desktop Linux distribution),Mac OS X 10.4.8 or later, Windows XP or Vista. Requirements also include Java Development Kit, Apache Ant, and Python 2.2or later. The officially supported integrated development environment (IDE) is Eclipse (3.2or later) using the Android Development Tools(ADT) Plugin, though developers may use any text editor to edit Java and XML files then use command line tools to create, build and debug Android applications.

In 26thOctober, 2009 the "Eclair" version (2.0)was released which featured better search, battery usage indicator and VPN control applet. New platform technologies included...

- Optimized hardware speed
- Support for more screen sizes and resolutions
- Revamped UI
- New browser UI and HTML5 support
- New contact lists
- Better white/black ratio for backgrounds
- Improved Google Maps 3.1.2
- Microsoft Exchange support
- Built in flash support for Camera
- Digital Zoom
- Improved virtual keyboard
- Bluetooth 2.1

Android Applications are package din .apk format and stored under /data/app folder on the Android OS. The user can run the command adb root to access this folder as only the root has permissions to access this folder.

2.2 ADT Plug-in:

The Android Development Tools (ADT) plug-in for Eclipse adds powerful extensions to the Eclipse integrated development environment. It allows you to create and debug Android applications easier and faster. If you use Eclipse, the ADT plugin gives you an incredible boost in developing Android applications:

- It gives you access to other Android development tools from inside the Eclipse IDE.
- For Example, ADT lets you access the many capabilities of the DDMS tool: take screenshots, manage port-forwarding, set breakpoints, and view thread and process information directly from Eclipse.

- It provides a New Project wizard, which helps you quickly create and set up all of the basic files you'll need for a new android application.
- It automates and simplifies the process of building your android application.
- It provides an android code editor that helps you write valid XML for your android manifest and resource files.
- It will even export your project into a signed APK, which can be distributed to users.

2.3 Emulator:

The Android SDK includes a mobile device emulator – a virtual mobile that runs on your computer. The emulator lets you prototype, develop, and test Android applications without using a physical device.

The android emulator mimics all of the typical hardware and software features of a typical mobile device, except that it cannot receive or place actual phone calls. It provides a variety of navigation and control keys, which you can "press" using your mouse or keyboard to generate events for your application. It also provides a screens in which your application is display, together with any other android applications running.

To let you model and test your application more easily, the emulator supports android virtual device (AVD) configurations. AVDs let you specify the android platform that you want to run on the emulator, as well as the hardware options and emulator skin files that you want to use. Once your application is running on the emulator, it can use the services of the android platform to invoke other applications, access the network, play audio and video, store and retrieve data, notify the user, and render graphical transitions and themes.

The emulator also includes a variety of debug capabilities such as console from which you can log kernel output, simulate application interrupts (such as arriving SMS messages or phone calls), and simulate latency effects and dropouts on the data channel.

2.4 Emulator Limitations:

No support for placing or receiving actual phone calls. You can simulate phone calls (placed and received) through the emulator console, however:

No support for USB connections

No support for camera/video capture (input).

No support for device-attached headphones.

No support for determining connected state.

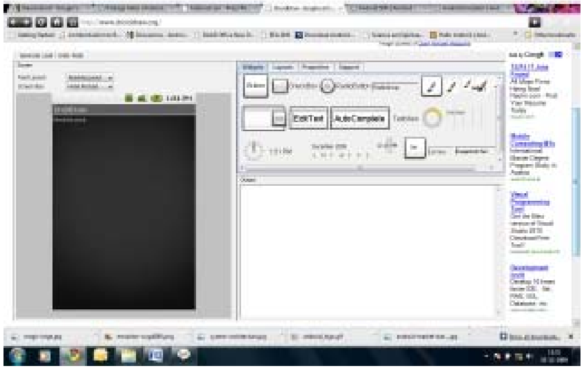
No support for determining battery charge level and AC charging state.

No support for determining SD card insert/eject.

No support for Bluetooth.

3 DRAWDROID:

It is a simply java application to design a GUI for any android application. It contains large number of widgets, layouts, buttons and all that we need to make a well fledge android app. Through drag and drop, we can design the layout and in back end it will generate the xml code which we can use directly in any particular application.



Interesting Apps in market:

1. ASE
2. Barcode scanner
3. Notepad

4. GPS status 2
5. I-Jetty
6. Nimbuzz
7. OI File manager
8. Opera mini browser
9. telnet
10. terminal emulator

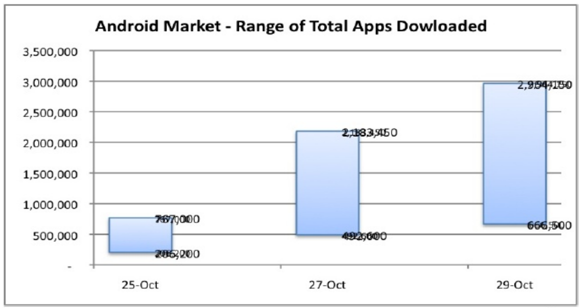
11. Application that we developed:

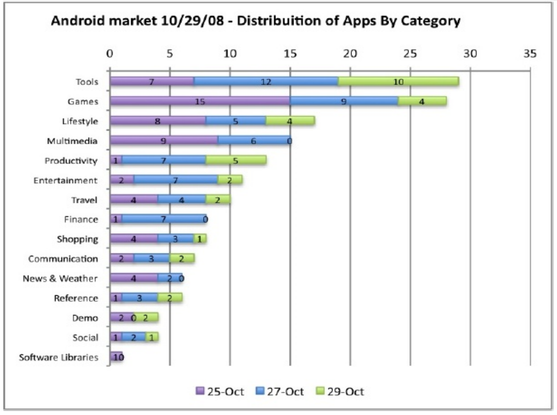
- **Text Animation:** Successfully made text animation that would OMIT the HTML tags and renders information without Coding in any form.
- **Calcy:** Developed a Mobile Calculator using Android. Can implement all Arithmetic functions which are possible with the modern World Scientific Calculator.
- **Brew clock:** Successfully created brew clock in android phone, which helps to count down the timer.

12. **Market Analysis:** While Google are keeping the exact download statistics private, mobile advertising network Media lets have been doing some data analysis of their own and have publicized some approximate figures and trends for the Android Market's first week end. Here a rundown of the statistics Media lets have published:

- 167 Apps have been downloaded between667,000 and 2.9 Million times.
- Downloads are being driven by 41 apps, which account for between 73% and 83%of all possible downloads.
- Two applications, Pac-Man by Namco and The Weather channel generated the most downloads (50,000 - 250,000).
- Although download ranges can be very large, growth on the low-end was still nearly 80% during the first week.
- 5 Categories account for 61% of all apps.
- The number of apps has nearly tripled since launch, led by the Games, Tools and Productivity categories.

4 SCREEN SHOTS





4.1 Future Ideas:

- **GPS Status:** Can be used for keeping a track of the Vehicles(Can be used by transport agencies),Can be used for keeping a track of trains, Military operations, Finding the shortest distance, Finding location on a Map. It can be integrated using Google Maps and can find use in a number of applications.

- **Browser for Data Encryption**: Could beused for Security Purposes CustomizedBrowserthat may render ApplicationSpecific information from a given codedHTML Light and faster browser which isthe need of the hour considering limitedresources in Mobile phones.

- **Embedding with Google wave:** TheInformation sharing that we are talking of deals with: Internships, ProjectAssociates, Sponsors, Guides, Colleges(their feedbacks and placements),Coaching, Schools, Exams, Results,Scholarships, Awards, Placements, JobProfiles, Job Feedbacks, CompetitiveExams and their Scope, Right approachtowards preparation, Counselling, GoodBooks and Study material etc.

4.2 Google Map

In order to use Google Maps in an Android application, you need to make sure that the application is properly setup to be able to host a Map View .Although it only takes a couple of steps to prepare your application for using Google Maps, failing to complete or skipping one of these steps can quickly turn this into an error-prone process.

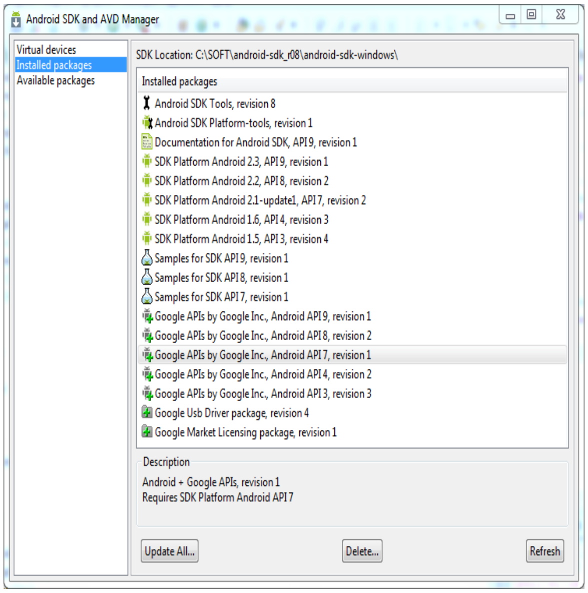
In short, the following needs to happen to enable your application for Google Maps :

- The **Android SDK** needs to be setup to support Google Maps.
- A Map View component needs to be added to your layout, containing a valid **Google Maps API key**
- The Activity that will be responsible for showing the map needs to extend from **Map Activity**
- The application manifest needs to be setup with the **android. permission. INTERNET** permission
- The application manifest needs to be setup with the **com. google. android. maps** library

4.3 Setting up our Android SDK for Google Maps

If you want to do anything with Google Maps, you'll need to ensure that your Android SDK has the necessary components installed. A key component is called the **Google APIs Add-on**, as according to Google is an extension to the Android SDK development environment that lets you develop applications for devices that include Google's set of custom applications, libraries, and services. A central feature of the add-on is the Maps external library, which lets you add powerful mapping capabilities to your Android application.

More information on how to install the Google APIs Add-on can be found here. After the installation, you should see the Google APIs in your installed packages.



Android SDK and AVD Manager

SDK Location: C:\SOFT\android-sdk_r08\android-sdk-windows\

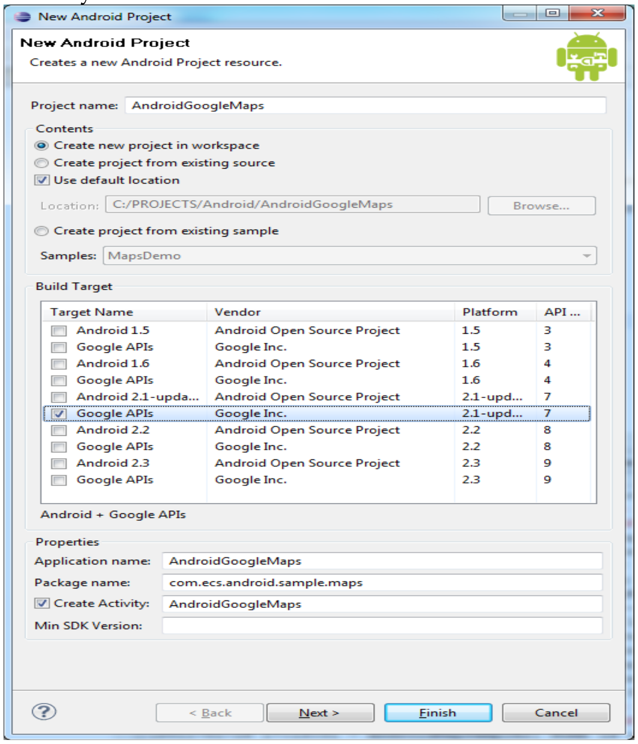
Installed packages

- Android SDK Tools, revision 8
- Android SDK Platform-tools, revision 1
- Documentation for Android SDK, API 9, revision 1
- SDK Platform Android 2.3, API 9, revision 1
- SDK Platform Android 2.2, API 8, revision 2
- SDK Platform Android 2.1-update1, API 7, revision 2
- SDK Platform Android 1.6, API 4, revision 3
- SDK Platform Android 1.5, API 3, revision 4
- Samples for SDK API 9, revision 1
- Samples for SDK API 8, revision 1
- Samples for SDK API 7, revision 1
- Google APIs by Google Inc., Android API 9, revision 1
- Google APIs by Google Inc., Android API 8, revision 2
- Google APIs by Google Inc., Android API 7, revision 1
- Google APIs by Google Inc., Android API 4, revision 2
- Google APIs by Google Inc., Android API 3, revision 3
- Google Usb Driver package, revision 4
- Google Market Licensing package, revision 1

Description

Android + Google APIs, revision 1
Requires SDK Platform Android API 7

Now that we have setup your Android SDK with the Google APIs, you can create Android Projects in Eclipse that target these Google APIs. This is required in order to be able to use the classes residing in the maps.jar file (classes like Map Activity



New Android Project

New Android Project
Creates a new Android Project resource.

Project name: AndroidGoogleMaps

Contents
● Create new project in workspace
○ Create project from existing source
☑ Use default location
Location: C:/PROJECTS/Android/AndroidGoogleMaps

○ Create project from existing sample
Samples: MapsDemo

Build Target

	Target Name	Vendor	Platform	API ...
□	Android 1.5	Android Open Source Project	1.5	3
□	Google APIs	Google Inc.	1.5	3
□	Android 1.6	Android Open Source Project	1.6	4
□	Google APIs	Google Inc.	1.6	4
□	Android 2.1-upda...	Android Open Source Project	2.1-upd...	7
☑	Google APIs	Google Inc.	2.1-upd...	7
□	Android 2.2	Android Open Source Project	2.2	8
□	Google APIs	Google Inc.	2.2	8
□	Android 2.3	Android Open Source Project	2.3	9
□	Google APIs	Google Inc.	2.3	9

Android + Google APIs

Properties
Application name: AndroidGoogleMaps
Package name: com.ecs.android.sample.maps
☑ Create Activity: AndroidGoogleMaps
Min SDK Version:

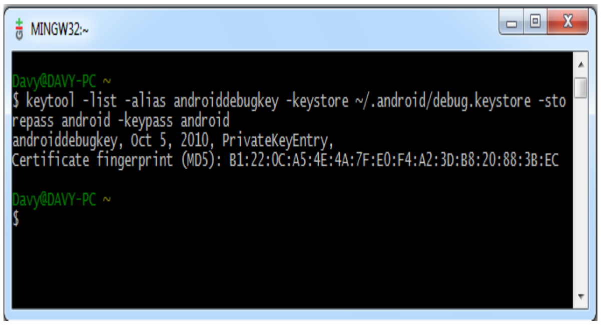
4.4 Retrieve a Google Maps API Key:

Before we can start using maps in our Android application, you need to obtain a Maps API Key. Signing up for such an API key is fairly straightforward, and requires you to do the following :

- Registering the MD5 fingerprint of the certificate that you will use to sign your application. The Maps registration service then provides you a Maps API Key that is associated with your application's signer certificate.
- Adding a reference to the Maps API Key in each Map View, whether declared in XML or instantiated directly from code. You can use the same Maps API Key for any Map View in any Android application, provided that the application is signed with the certificate whose fingerprint you registered with the service.

More information on getting a Map Key for development purposes can be found here.

You'll need to run a command line tool called key tool to display the certificate fingerprint. The key tool command is distributed with any Java JDK / JRE and can be located under %JAVA_HOME%/bin



```
Davy@DAVY-PC ~
$ keytool -list -alias androiddebugkey -keystore ~/.android/debug.keystore -sto
repass android -keypass android
androiddebugkey, Oct 5, 2010, PrivateKeyEntry,
Certificate fingerprint (MD5): B1:22:0C:A5:4E:4A:7F:E0:F4:A2:3D:B8:20:88:3B:EC

Davy@DAVY-PC ~
$
```

Once we have the API key, we can add our Map View component in the main layout like this :

Important : Your API key is linked to the Android SDK key store you're currently using. This means that if you run your application on another development environment, where a different Android SDK key store is installed, the API key will not work, and you won't see any tiles being loaded on the Map.

Setting up our emulator for Google Maps*:* If you want to test your application through the emulator, you'll need to have an AVD (Android Virtual Device) that has includes the Google APIs Add-On. Before you will be able to create such an AVD, you need to ensure that you've installed the Google APIs Add-On.

When creating an emulator, make sure it has the proper target defined. The target should be called something like Google APIs (Google Inc).

5. CONCLUSION

5.1 **Conclusion:** This paper helps us to view the games. We created project in a user friendly manner.. I hope this project will be very helpful for the end users. This paper presents a system model that can apply to our management. It is made up of three layers user layer, application layer and resource layer. In this paper mobile knowledge service of complex project is summarized into two service mode which includes mobile knowledge search server and mobile knowledge management server.

5.2 **Future enhancement**: The project can be upgraded by providing more features and user friendly for the user:-can be extended to many platforms to integrate huge interface using advanced applications and its tools.

By configuring our server with commercial sites.
This project can be upgraded by using high end application.
This project can also be used for high end server crash monitoring and configuring
Server database from remote places.

6 *REFERENCES:*

[1] http://developer.android.com
[2] http://wikipedia.com
[3] http://android.com/market
[4] http://androidapps.com