© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 06 Issue: 02 | May-June 2015

Study of Android application Development using various tools

Akash Ahlawat, MTECH 4th sem, department of computer science and engineering
SBMN engineering college, MD university, Rohtak

Abstract: *Android is an open source and Linux-based **Operating System** for mobile devices such as smart phones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.*

Keywords: Cloud Computing, Jelly Beans, Android Development Kit, .Net Framework

I. Introduction

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android. The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 **Jelly Bean**. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

Fig 1

II. Features of Android

Android is a powerful operating system competing with Apple 4GS and supports great features. Few of them are listed below:

Feature	Description
Beautiful UI	Android OS basic screen provides a beautiful and intuitive user interface.
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Storage	SQLite, a lightweight relational database, is used for data storage purposes.
Media support	H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP
Messaging	SMS and MMS
Web browser	Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
Multi-tasking	User can jump from one task to another and same time various application can run simultaneously.
Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space
Multi-Language	Supports single direction and bi-directional text.

GCM	Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.
Wi-Fi Direct	A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
Android Beam	A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

Table 1

III. Android Applications

Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play**, **SlideME**,**O pera Mobile Store**,**Mobango**, **F-droid** and the **Amazon Appstore**.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

IV. Categories of Android applications

There are many android applications in the market. The top categories are:

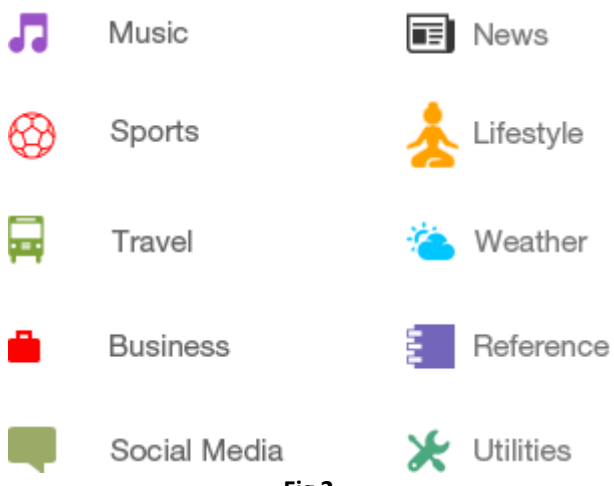

Fig 2

V. History of Android

The code names of android ranges from A to L currently, such as Aestro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwitch, Jelly Bean, KitKat and Lollipop.

Fig 3

VI. Android Application Development Tools

We can start your Android application development on either of the following operating systems −

- Microsoft Windows XP or later version.
- Mac OS X 10.5.8 or later version with Intel chip.
- Linux including GNU C Library 2.7 or later.

Second point is that all the required tools to develop Android applications are freely available and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

- Java JDK5 or later version
- Android SDK
- Java Runtime Environment (JRE) 6
- Android Studio
- Eclipse IDE for Java Developers (optional)
- Android Development Tools (ADT) Eclipse Plug-in (optional)

Here last two components are optional and if you are working on Windows machine then these components make your life easy while doing Java based application development. So let us have a look how to proceed to set required environment.

Set-up Java Development Kit (JDK) for android programming

We can download the latest version of Java JDK from Oracle's Java site: Java SE Downloads. We will find instructions for installing JDK in downloaded files, follow the given instructions to install and configure the setup. Finally set PATH and JAVA_HOME environment variables to refer to the directory that contains **java** and **javac**, typically java_install_dir/bin and java_install_dir respectively.

If you are running Windows and installed the JDK in C:\jdk1.6.0_15, you would have to put the following line in your C:\autoexec.bat file.

set PATH=C:\jdk1.7.0_75\bin;%PATH%
set JAVA_HOME=C:\jdk1.7.0_75

Alternatively, you could also right-click on *My Computer*, select *Properties*, then *Advanced*, then *Environment Variables*. Then, you would update the PATH value and press the OK button.

On Linux, if the SDK is installed in /usr/local/jdk1.6.0_15 and you use the C shell, you would put the following code into your **.cshrc** file.

setenv PATH /usr/local/jdk1.7.0_75/bin:$PATH
setenv JAVA_HOME /usr/local/jdk1.7.0_75

Alternatively, if you use an Integrated Development Environment (IDE) Eclipse, then it will know automatically where you have installed your Java.

Android IDEs

There are so many sophisticated Technologies are available to develop android applications, the familiar technologies, which are predominantly using tools as follows

- Android Studio
- Eclipse IDE

User Interface Layout

The basic building block for user interface is a **View** object which is created from the View class and occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components like buttons, text fields, etc.

The **ViewGroup** is a subclass of **View** and provides invisible container that hold other Views or other ViewGroups and define their layout properties.

At third level we have different layouts which are subclasses of ViewGroup class and a typical layout defines the visual structure for an Android user interface and can be created either at run time using **View/ViewGroup** objects or you can declare your layout using simple XML file **main_layout.xml** which is located in the res/layout folder of your project.

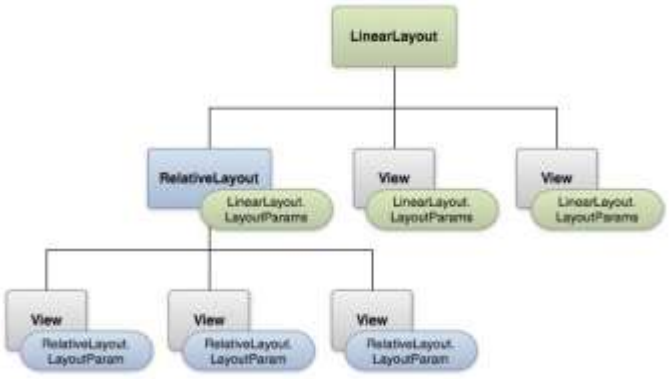

Fig 4

Android programming on .Net Platform

In many ways, the Android is an easy catch for .NET developers. First and foremost, the programming language is either C#, if you opt for writing apps through the Mono platform, or Java-a close relative to C#-if you take the classic route of the standard Android SDK.

Second, you don't need to buy a Mac in order to develop for Android. Whatever IDE you choose to use is available for Windows too. Finally, the user interface of Android apps is laid out using an XML schema that doesn't look foreign to people who know XAML.

Compared to developing for iOS or even Windows Phone, Android programming is challenging for one notable reason: the huge number of significantly different devices that may be running your application. You can find Android on an $80 cell phone as well as on a high-end $500 smartphone.

The difference in terms of power, battery, resources, and screen size can be huge. These variables add more work to the schedule and, even more than that, it adds forethought: you must plan your Android effort well for it to result in a success story.

Options for Android Development on .Net

The primary approach to Android programming consists of downloading the Android SDK and writing your apps using the Java programming language. If you're a .NET developer, though, you might want to consider another option: Xamarin Studio from Xamarin. Note that Xamarin Studio (formerly MonoDroid) is a commercial product and may require you to buy a license. (For more information, have a look at http://www.xamarin.com.)

You can use the Xamarin Studio IDE (formerly MonoDevelop) on a Windows computer or you can install an extension to Visual Studio and keep using all of your favorite plug-ins.

An Android application built with Xamarin Studio executes within an instance of a Mono-based virtual computer. The Mono virtual computer, in turn, lives side-by-side with the Android's virtual computer. To access native Android functionalities, you use a bunch of classes that look like classes in the .NET Framework except that they bind to the Android API under the hood.

The build of a Mono-based Android application passes through four steps: processing resources from Android resource files into .NET-compatible resource files; creation of the .NET code; processing of the .NET code to create Java wrappers; and final packaging of the Android executable.

Another option for writing Android applications is PhoneGap (see http://phonegap.com.) An app built against the PhoneGap framework is a classic client-side Web application made of HTML5 pages using CSS and JavaScript. You write and test the application within your favorite development platform, and using your favorite tools. For example, you can use Visual Studio or perhaps JetBrains's WebStorm or any other text editor for the HTML pages and JavaScript logic. Once the Web core is ready, though, you have the problem of packaging that into a real Android application. You need an IDE for that like Eclipse or JetBrains's IntelliJ IDEA. eOr, you can leverage the PhoneGap Build service (http://build.phonegap.com) and upload your app to the service for packaging.

VII. Reference

[1] David Chappell, a Short Introduction to Cloud Platform; An Enterprise - Oriented Overview (2008)

[2] Introduction to Cloud Computing architecture White Paper 1st Edition, June 200

[3] Prime Minister's official residence: i-Japan strategy 2015, (in Japanese),

[4]http://www.kantei.go.jp/jp/singi/it2/kongo/digital/dai9/9siryou2.pdf Ministry of Internal Affairs and Communications

[5] Windows Azure, http://msdn.microsoft.com/jajp/azure/cc994380.aspx

[6] Google Apps ,http://www.google.com/a/help/intl/ja/admins/customers.html

[7] Amazon Elastic Compute Cloud (Amazon EC2) http://aws.amazon.com/ec2/#pricing

[8] M.V. Luis, R. M. Luis, C. Juan, L. Maik.

[9] A Break in the Clouds: Towards a Cloud Definition. Computer Communication Review, vol.39, pp.50-55, 2009

[10]OL.Googlehttp://developer.android.com/guide/topics/fundamentals.html

[11] M. Fengsheng Yang, Android Application Development Revelation, China Machine Press, 2010.

[12] J. Li Lin, ChangweiZou, Research on Cloud Computing Based on Android Platform, vol.11. Software Guide, 2010, pp.137-139

[13] Jianye Liu, Jiankun Yu, Research on Development of Android Applications, IEEE computer society, 2011

[14]http://www.tutorialspoint.com/android/index.htm