ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

Design and Study of Android Based SDK'S for Mobile Devices

Dr. G. Shankar Lingam[1]

Professor, Dept. of CSE, Chaitanya Institute of Technology & Science, Warangal, Telangana, India[1]

ABSTRACT: Modern hand-held devices which includes smart phones and PDAs have turn out to be more and more effective in recent years.Dramatic breakthroughs in processing energy in conjunction with the wide variety of greater features protected in these devices haveopened the doors to an extensive range of commercial possibilities. In precise, most mobile phones often encompasscameras, processors similar to PCs from just a few years ago, and net acces.This paper describes improvement of software on Android mobile platform. This paper focuses mostly on the Android structure which is based totally on Linux model 2.6. It is an open-source mobile smartphone operating system which is Linux-primarily based. Android packages are written in Java programming language. Android SDK gives set of application programming interfaces (APIs) and Eclipse Platform that can be used to create programs.

KEYWORDS: Android limitation, Application behavior, Android SDK Manager

I. INTRODUCTION

In current years, the emergence of smart phones has modified the definition of mobile phones. Phone is not only a communication device, however also a vital a part of the humanbeing's communication and day by day life. Various packages brought unlimited a laugh for human's lives. It is sure that the destiny of the network could be the mobile terminal.Now the Android device within the electronics marketplace is becoming increasingly more famous, mainly inside the telephone marketplace. Because of the open source, a number of the development tools are loose, so there are masses of programs generated. This greatly inspired the human beings to apply the Android device. In addition, it affords a totally convenient hardware platform for builders so we can spend less effort to comprehend their thoughts. This makes Android can get further development [1-4].
As the smart telephones and Android machine getting famous, the operations like listening to track, watching videos, tweeting and some others can be moved from the computer to a smartphone now.The applications on the market nowadays are basically business applications, and incorporate a massive wide variety of integrated advertising and marketing. If the user prefers to get rid of the integrated advertising, sure charges have to be paid to reach that and this is not handy. Meanwhile, because of the unfair competition of IT, many programs constructed unlawful software to steal person information and cause some harm to user's private privacy. Sometimes, users will pay greater attention to the user experience of software program. Therefore, the improvement of the application can't handiest be restrained to the function, greater interest should be paid to the user's enjoy. After reading a few preceding Android programs and get admission to massive amounts of materials, we utilize the Java language, the Eclipse platform, Android ADT and the Android SDK to develop these three mobile programs. These structures have a pleasing interface and smooth operation. These Apps won't steal any private records, but can exclude useless statistics and convey a notable user experience.

II. RELATED RELEASES IN ANDROID

Android is a Linux-based operating system for mobiledevices such as smart phones and tablet computers. Itis developed by the Open Handset Alliance led byGoogle. The android SDK provides the tools and APIsnecessary to begin developing applications on theandroid platform using the Java programminglanguage.
Recent releases:
1: Gingerbread refined the user interface, improvedthe soft keyboard and copy/paste features, better nativecode support (which improves gaming performance),added SIP support (VoIP calls), and added support forNear Field Communication.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

2: Honeycomb was a tablet-oriented release whichsupports larger screen devices and introduces manynew user interface features, support for multicoreprocessors, hardware acceleration for graphics and fullsystem encryption. The first device featuring thisversion, the Motorola Xoom tablet, went on sale inFebruary 2011.
3: Honeycomb, released in May 2011, and addedsupport for extra input devices, USB host mode fortransferring information directly from cameras andother devices, and the Google Movies and Books apps.
4: Honeycomb, released in July 2011, addedoptimization for a broader range of screen sizes, new"zoom-to-fill" screen compatibility mode, loadingmedia files directly from SD card, and an extendedscreen support API. Huawei Media Pad is the first 7inch tablet to use this version.
5: 4.0 Ice Cream Sandwich, announced on October19, 2011, brought Honeycomb features to smart phonesand added new features including facial recognitionunlock, network data usage monitoring and control,unified social networking contacts, photographyenhancements, offline email searching, app folders, andinformation sharing using NFC. Android 4.0.4 IceCream Sandwich is the latest Android version that isavailable to phones. The source code of Android 4.0.1was released on November 14, 2011

III. SYSTEM OVERVIEW

Android Development Tool plugins for EclipseAndroid Development Tools (ADT) is a plugin for theEclipse IDE that is designed to give you a powerful,integrated environment in which to build Androidapplications. ADT extends the capabilities of Eclipse tolet us quickly set up new Android projects, create anapplication UI, add packages based on the AndroidFramework API, debug your applications using theAndroid SDK tools, and even export signed (orunsigned) .apk files in order to distribute ourapplication. Developing in Eclipse with ADT is highlyrecommended and is the fastest way to get started.
With the guided project setup it provides, as well astools integration, custom XML editors, and debugoutput pane, ADT gives we an incredible boost indeveloping Android applications. Many of the toolsthat we can start or run from the command line areintegrated into ADT. They include:

- **Traceview:** Allows us to profile ourprogram's execution (Window > OpenPerspective >Traceview).

- **Android:** Provides access to the AndroidSDK Manager and AVD Manager. Otherandroid features such as creating or updatingprojects (application and library) areintegrated throughout the Eclipse IDE.

- **Hierarchy Viewer:** Allows us to visualizeour application's view hierarchy to findinefficiencies (Window > Open Perspective> Hierarchy Viewer).

- **Pixel Perfect:** Allows us to closely examineour UI to help with designing and building.(Window > Open Perspective > PixelPerfect).

- **DDMS:** Provides debugging featuresincluding: screen capturing, thread and heapinformation, and logcat (Window > OpenPerspective > DDMS).

- **adb:** Provides access to a device from ourdevelopment system. Some features of adb areintegrated into ADT such as projectinstallation (Eclipse run menu), file transfer,device enumeration, and logcat (DDMS). Wemust access the more advanced features ofadb, such as shell commands, from thecommand line.

- **ProGuard:** Allows code obfuscation,shrinking, and optimization. ADT integratesProGuard as part of the build, if we enable it.

- **Android SDK Tools:**Contains tools for debugging and testing yourapplication and other utility tools. These toolsare installed with the Android SDK starterpackage and receive periodic updates. We canaccess these tools in the <sdk>/tools/ directoryof our SDK. To learn more about them, seeSDK Tools in the developer guide.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2016

SDK Manager
The Android SDK Manager is the tool that you use toinstall and upgrade SDK packages in our developmentenvironment. We can launch the Android SDKManager in one of the following ways.
AVD Manager
An Android Virtual Device (AVD) is an emulatorconfiguration that lets our model an actual device bydefining hardware and software options to be emulatedby the Android Emulator. The easiest way to create anAVD is to use the graphical AVD Manager, which welaunch from Eclipse by clicking Window >AVDManager. We can also start the AVD Manager fromthe Command line by calling the android tool with theavd options, from the<sdk>/tools/ directory.
We can also create AVDs on the command line bypassing the android tool options. For more informationon how to create AVDs in this manner, see ManagingVirtual Devices from the Command Line.
An AVD consists of:
 A hardware profile: Defines the hardwarefeatures of the virtual device. For example, we can define whether the device has acamera, whether it uses a physical QWERTYkeyboard or a dialing pad, how much memoryit has, and so on.
 A mapping to a system image: We can definewhat version of the Android platform will runon the virtual device. We can choose a versionof the standard Android platform or thesystem image packaged with an SDK add-on.
 Other options: We can specify the emulatorskin we want to use with the AVD, which letswe control the screen dimensions, appearance,and so on. We can also specify the emulatedSD card to use with the AVD.
 A dedicated storage area on our developmentmachine: the device's user data (installedapplications, settings, and so on) and emulatedSD card are stored in this area.We can create as many AVDs as we need, basedon the types of device we want to model. Tothoroughly test our application, we should createan AVD for each general device configuration (forexample, different screen sizes and platformversions) with which our application is compatibleand test our application on each one. Keep thesepoints in mind when we are selecting a systemimage target for our AVD.

- The API Level of the target is important,because our application will not be able to runon a system image whose API Level is lessthan that required by our application, asspecified in the minSdk Version attribute ofthe application's manifest file. For moreinformation about the relationship betweensystem API Level and application minSdkVersion, see Specifying Minimum SystemAPI Version.
- We should create at least one AVD that uses atarget whose API Level is greater than thatrequired by our application, because it allowswe to test the forward-compatibility of ourapplication. Forward-compatibility testingensures that, when users who havedownloaded your application receive a systemupdate, our application will continue tofunction normally.
- If our application declares a uses-libraryelement in its manifest file, the application canonly run on a system image in which thatexternal library is present. If we want to runour application on an emulator, create an

AVD that includes the required library.Usually, we must create such an AVD usingan Add-on component for the AVD's platform(for example, the Google APIs Add-oncontains the Google Maps library). To learnhow to manage AVDs using a graphical tool,read Managing AVDs with AVD Manager. Tolearn how to manage AVDs on the commandline, read Managing AVDs from theCommand Line.

Android SDK Platforms (2.3.3 API Level 10GingerBread)
The platform tools are typically updated every time weinstall a new SDK platform. Each update of theplatform tools is backward compatible with olderplatforms. Usually, we directly use only one of theplatform tools— the Android Debug Bridge (adb).Android Debug Bridge is a versatile tool that lets wemanage the state of an emulator instance or Android powered device. We can also use it to install anAndroid application (.apk) file on a device. The otherplatform tools, such as aidl, aapt, dexdump, and dx, aretypically called by the Android build tools or AndroidDevelopment Tools (ADT), so we rarely need toinvoke these tools directly. As a general rule, weshould rely on the build tools or the ADT plugin to callthem as needed.
By default, there are two repositories of packages forour SDK: Android Repository and Third party Addons.
The Android Repository offers these types of packages:
 SDK Tools — Contains tools for debuggingand testing our application and other utilitytools. These tools are installed with theAndroid SDK starter package and receiveperiodic updates. We can access these tools inthe <sdk>/tools/ directory of our SDK. Tolearn more about them, see SDK Tools in thedeveloper guide.

 SDK Platform-tools — Contains platformdependent tools for developing and debuggingour application. These tools support the latestfeatures of the Android platform and aretypically updated only when a new platformbecomes available. We can access these toolsin the <sdk>/platform-tools/ directory. Tolearn more about them, see Platform Tools inthe developer guide.
 Android platforms — An SDK platform isavailable for every production Androidplatform deployable to Android-powereddevices. Each SDK platform package includesa fully compliant Android library, systemimage, sample code, and emulator skins. Tolearn more about a specific platform, see thelist of platforms that appears under the section"Downloadable SDK Packages" on the leftpart of this page.
 USB Driver for Windows (Windows only)— Contains driver files that we can install onour Windows computer, so that we can runand debug our applications on an actualdevice we do not need the USB driver unlesswe plan to debug your application on an actualAndroid-powered device. If we develop onMac OS X or Linux, we do not need a specialdriver to debug our application on anAndroid-powered device. See UsingHardware Devices for more information aboutdeveloping on a real device.
 Samples — Contains the sample code andapps available for each Android developmentplatform. If we are just getting started withAndroid development, make sure to downloadthe samples to our SDK.

V. CONCLUSION

According to our study from findings the Android Mobile Application Development is based on Javalanguage codes. It allows developers to write codes in the Javalanguage. These codes can control mobile devices via GoogleenabledJava libraries. It provides the platform to developmobile applications using the software stack provided in theGoogle Android SDK. Android mobile OS provides a flexibleenvironment for Android Mobile Application Development asthe developers can not only make use of Android JavaLibraries but it is also possible to use Java IDEs. The softwaredeveloper in Mobile Development has expertise in developingapplications based on Android Java Libraries and other important tools.

REFERENCES

1. M. Butler, “Android: Changing the Mobile Landscape”, Pervasive Computing, (2011), pp. 4-7.
2. B. Proffitt, “Open Android-For better and for worse”, Spectrum, (2011), pp. 22– 24.
3. K. W. Tracy, “Mobile Application Development Experiences on Apple’s iOS and Android OS”, Potentials,(2012), pp. 30 – 34.
4. A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, S. Dolev and C. Glezer, “Google Android: A Comprehensive Security Assessment”, Security & Privacy, (2010), pp. 35 – 44.
5. Location Manager
6. http://developer.android.com/reference/android/location /LocationManager.html
7. Location
8. http://developer.android.com/reference/android/location/Location.html
9. Shared Preferences
10. http://developer.android.com/reference/android/content/SharedPreferences.html
11. Location Listener
12. http://developer.android.com/reference/android/locatio n/LocationListener.html
13. Shared Preferences Editor
14. http://developer.android.com/reference/android/content/SharedReference.Editor.html
15. Broadcast Receiver
16. http://developer.android.com/reference/android/content/BroadcastReceiver.html
17. Sms Manager
18. http://developer.android.com/referece/android/telephony/SmsManager.html
19. Sms Message
20. http://developer.android.com/reference/android/telephony/SmsMessage.html
21. Audio Manager
22. http://developer.android.com/reference/android/media/AudioManager.html
23. Media Player
24. http://developer.android.com/reference/android/media/MediaPlayer.html

BIOGRAPHY

Dr. G. Shankar Lingam completed his MCA in Chaitanya Degree & P.G College and M.Tech in CSE from Ramappa Engineering College respectively. He is having teaching experience of more than 20 years in various Under Graduate and Post Graduate courses. He has guided lots of students in various Under Graduate and Post Graduate Research Projects. At Present, he is working Professor, Dept. of CSE, Chaitanya Institute of Technology & Science, Warangal, Telangana, India.