Android Application Development using Android Studio and PHP Framework

Akshay Singh
IMS Engineering College

Sakshi Sharma
IMS Engineering College

Shashwat Singh
IMS Engineering College

ABSTRACT

Today, as we all know that the development of hardware for mobile device is getting better and the performance index is veryhigh than the actual requirements ofthe software configuration. Phone's features are now more dependent on software or application. This paper describes development of Android mobile platform application. Development environment of Windows Mobile and Apple's iPhone are very simplified for mobile applications. Mobile applications give users a quick and reliable user experience. Primary focus of this paper is on the Android architecture based on Linux version 2.6. It is Linux based an open-source mobile phone operating system. Basically Java programming language is used to develop android application. Android SDK provides set of application programming interfaces (APIs) and Android Studio Platform that can be used to create applications.

Keywords

Android, application framework, Android SDK, Android Studio.

1. INTRODUCTION

In recent years, the definition of mobile phones has been changed by the emergence of smart phones. It is no longer just a communication tool, but also become an essential part of the people's daily life. Various applications of android provides fun and other needs of daily life. It provides a medium to connect with other people.

This era is very great and exiting for mobile developers. Android [2] is an open source architecture that includes the Operating system, application framework, Linux kernel, middleware and application along with a set of API libraries for writing mobile applications that can give look, feel, and function of mobile handsets. Mobile developers can now expand into the Android platform to enhance reliability, usability and other features of existing products. Without any trouble and complexity, Android developers can easily write application code that can make mobile hardware more useful and user friendly. In global mobile market, mobile applications are a rapidly growing. In this paper, we discuss on Android Studio for the mobile application development, Laravel (PHP Framework), layered approach for android, MySQL for database design and development. Android is an open-source mobile phone operating system based on Linux and developed by Google[2] and now a day it becomes the most widely used Operating System on mobile phones. Java language codes are used to develop android mobile application which allows developers to write codes in the Java language. Using Google enabled Java libraries these codes can control mobile devices. Android mobile Operating System also provides a flexible environment for development of Android Mobile Application as the developers can not only make use of Java IDEs but it is also possible to use Android Java Libraries. Android Mobile Application Development is basically used to create user friendly and innovative applications. Mobile Development has worked extensively on projects like video and audio players, gaming software, photo viewer and editor and more.

2. ANDROID ARCHITECTURE [3]

We studied the Android operating system architecture. Android system is a Linux-based system, Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.Each layer of the lower encapsulation, while providing call interface to the upper.

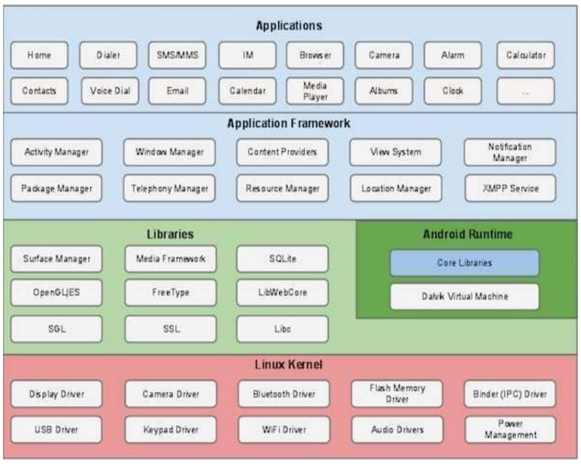


Figure 1. Android Architecture

2.1. Applications

At the top layer you will find all the Android application. If you want to install your own application or if you want to write your own application code then you can do it only on this layer. Examples of such applications are Location share, Browser, Share Application and Games etc.

2.2. Application Framework

All the API framework of the core programs can be easily accessed by developers and they can also modify it according to their need. The application framework make easier to understand the reuse of its components. If you want to access the functional components of other Android Application then you have to check whether the Application has release itor not and you must have to follow the security of the framework. Same as the users can be able to replace the program elements with this reuse mechanism.

2.3. Libraries and Android Runtime

The library is divided in to two main elements: Android Runtime and Android Library.

Android Runtime is made up of a Core Java Libraries and Dalvik virtual machine (The **Dalvik VM** executes files in the **Dalvik** Executable (.dex) format which is optimized for minimal memory footprint). Most functions of the core java

libraries is provided by Core library. DVM is register virtual machine and makes some specific developments for mobile devices.
Android system library is reinforce the application framework, it is also amain connection linking between application framework and Linux Kernel. This system library is enlarged in C or C++ language. These libraries can also be used by the different elements in the Android system. Service are provided to the developers through the application framework.

2.4. Linux Kernel

Last layer of the android architecture Linux - Linux 3.6 with approx. 115 patches. This gives a level of speculation between the device hardware and it carries all the required hardware drivers like display, camera, keypad etc. Also, the kernel take care of all the things that Linux is great at such as networking and a huge array of device drivers, which take the burden out of interfacing to peripheral hardware. The kernel system provides the operations like internal storage, process management, internet protocol, bottom-drive and other core service are all based on Linux kernel.

3. BACKGROUND

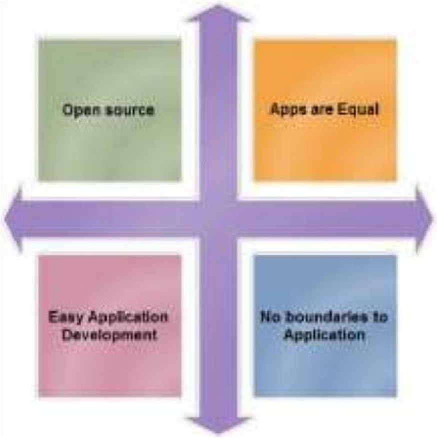


Fig. 1 Four core features of the android platform

Android is an open source Operating System for Mobile devices, It is initially developed by Android, Inc., which was bought by Google in 2005, Android was revealed in 2007, along with the founding of the Open Handset Alliance – a consortium of software,hardware and telecommunication companies dedicated to advancing open standards for mobile devices. According to the Wikipedia in July 2013, there were over one million Android application were published on the Google Play store, and over 50 billion applications downloaded. According tothe Wikipedia in April–May 2013 survey of mobile application developers found that more than 71% of developers created applications for Android.

3.1. Android Version History

Android mobile operating system has begun its version history with the release of the Android beta version in November 2007. Android 1.0 (First version),the first commercial version was released in September 2008. As we all know that Android is introduce byGoogleand theOpen Handset Alliance(OHA), and since its initial release, we have seen a number ofupdatesto its base operating system.

- Alpha (1.0)
- Beta (1.1)
- Cupcake (1.5)
- Donut (1.6)
- Éclair (2.0–2.1)
- Froyo (2.2–2.2.3)
- Gingerbread (2.3–2.3.7)
- Honeycomb (3.0–3.2.6)
- Ice Cream Sandwich(4.0–4.0.4)
- Jelly Bean(4.1–4.3.1)
- KitKat (4.4–4.4.4)
- Lollipop(5.0)
- Marshmallow (6.0)

3.2. Android Application development Implementation

3.2.1 Prototyping

Prototyping is the important part of the app which decides how the app look like after implementation. Photoshop CS6 has been used for prototyping of the Android App UI design.

3.2.2 UI design in XML View of the app – Implementation of View

Material design has been used with the latest ripple effect and flat UI with fragments in the app. The fragments gives more control of the element instead of simple activity alone.

3.2.3 Testing of UI in different versions of android using Genymotion

Maximum backwards compatibility of the app been set to Ice cream sandwich (Android 4.0.3) and maximum up to Marshmallow (Android 6.0) and recently has been updated to Android N preview version.

3.2.4 Implementation of Singleton class Of Volley Instantiation

Volley library has been used for transferring the data from the server and to the server asynchronously so that the app does not hang up working network task in the main thread. The singleton class keeps track of all the requests in the form of queue.

3.2.5 Parsing JSON data from the server – Implementation of Models

The JSON data from each call from the server is first parsed to store in the Saved Instances which is kind of local cache for the data to increase the performance of the app with long UI list.

3.2.6 Implementation of the Controller – The Adapter Class

The adapter class in the app does the exact work as the controller does. It transfers the request from the fragment or activity to the volley and then updates back the fragment using asynchronous calls from the model.

3.2.7 Handling the Images – Picasso Library

All the images in the app are loaded asynchronously using Picasso library. The library uses local caching to persist the large image sets in the app itself.

3.2.8 Using Git for Version Control

All the code revisions are saved in the form of version so that in case of any disaster like accidently overriding the working code.

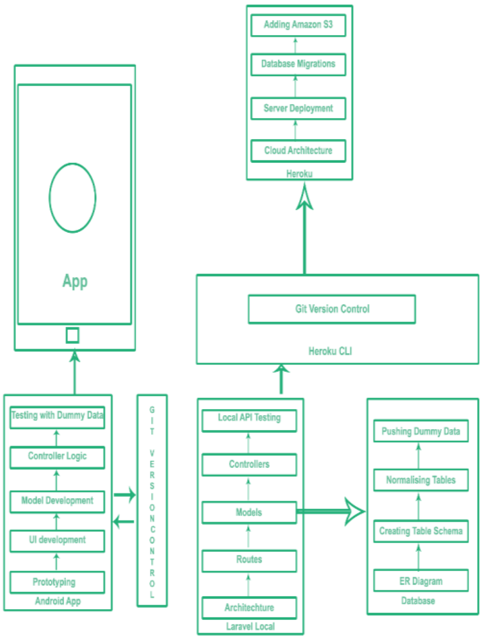


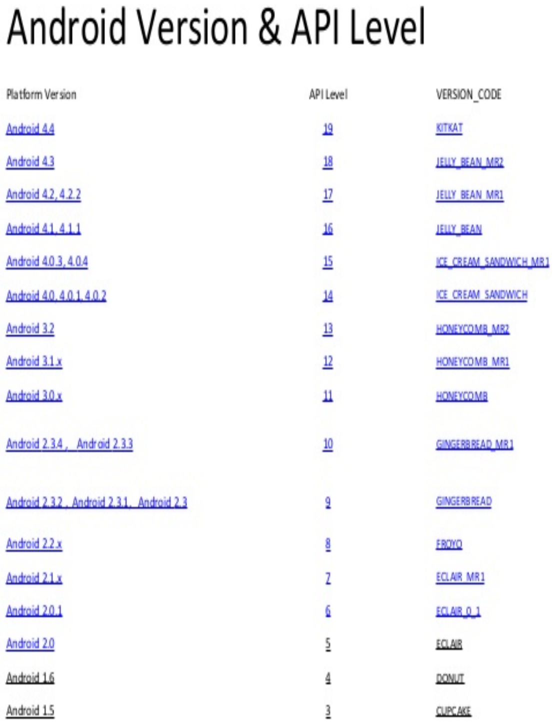
Fig. 2 Design Diagram

3.3. API (Application Program Interface)

API Level [6] is an integer value that incomparably recognizes the framework API revision provided by a version of the Android platform. The Android platform offers so many different framework API that applications can easily use to interact with the hidden Android system. We have specified the API Level supported by each version of the Android platform in table 1. API is application programming interface for other web or app based technologies. The API should be independent of the platform so that every platform could leverage the API calls according to their needs. In this case, API gives the output in JSON format so any platform supporting the Http connection can call to this API and develop their platform. Since the API development can be used by any application so the code should be as generic as possible.

API can be developed using many technologies available but due to the popularity and open source nature, PHP has been used. Although the API code can be written in PHP core alone but there are many security issues and reinventing the wheels kind of things to do in PHP core. So going with the framework is the best option to choose from for API development.

Table 1: API level [3]



Android Version & API Level

Platform Version	API Level	VERSION_CODE
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 , Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 , Android 2.3.1, Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE

3.4. Selecting PHP framework

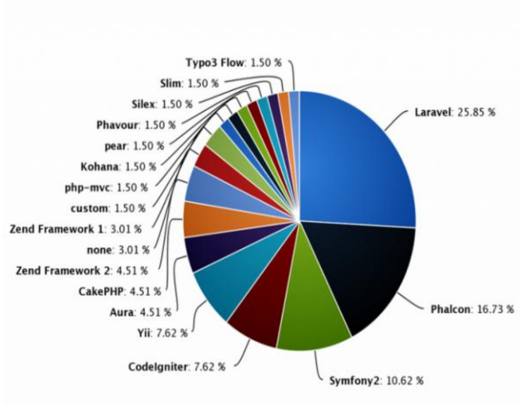


Fig. 3 PHP Framework

Following are the reasons for selecting the Laravel framework

- Popularity
- Community Support
- Security
- Backed with Zend Foundation which is the maintained by IBM and Google
- OAuth 2.0 Implementation
- Multiple Database Support
- Command Line Tools
- Strict MVC Based framework
- PHP Auto load Feature
- Composer Dependency Manager

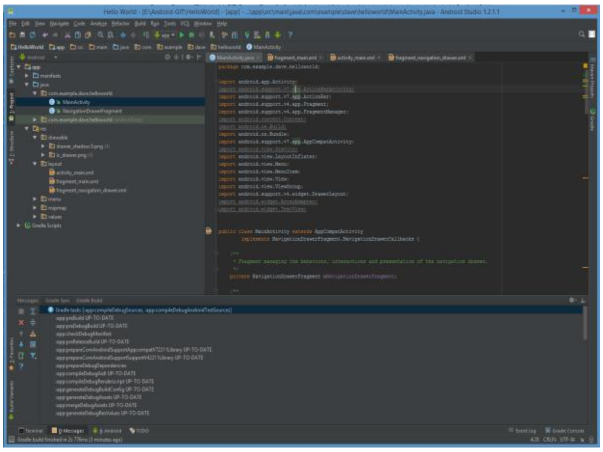
- Modular Approach
- Less Learning Curve

3.5. Android Studio [11]

Android Studio is the official integrated development environment (IDE) for Android platform development. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Based on Jet Brains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

Table 2 Android studio vs. Eclipse

Feature	**Android Studio**	**Eclipse ADT**
Build system	Gradle	Apache Ant
Maven-based build dependencies	Yes	No
Build variants and multiple-APK generation	Yes	No
Advanced Android code completion and refactoring	Yes	No
Graphical layout editor	Yes	Yes
APK signing and keystore management	Yes	Yes
NDK support	Yes	Yes



Hello World - [E:\Android-GIT\HelloWorld] - [app] - ...\app\src\main\java\com\example\dave\helloworld\MainActivity.java - Android Studio 1.2.1.1
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
HelloWorld app src main java com example dave helloworld MainActivity
Android
app
manifests
java
com.example.dave.helloworld
MainActivity
NavigationDrawerFragment
res
drawable
menu
mipmap
values
Gradle Scripts
MainActivity.java
fragment_main.xml
activity_main.xml
fragment_navigation_drawer.xml
Messages
Gradle Sync
Gradle Build
Terminal
0: Messages
6: Android
TODO
Event Log
Gradle Console

```
package com.example.dave.helloworld;

import android.app.Activity;
import android.support.v7.app.ActionBarActivity;
import android.support.v7.app.ActionBar;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.support.v4.widget.DrawerLayout;
import android.widget.ArrayAdapter;
import android.widget.TextView;


public class MainActivity extends AppCompatActivity
        implements NavigationDrawerFragment.NavigationDrawerCallbacks {

    /**
     * Fragment managing the behaviors, interactions and presentation of the navigation drawer.
     */
    private NavigationDrawerFragment mNavigationDrawerFragment;

    /**
```

```
Gradle tasks [app:compileDebugSources, app:compileDebugAndroidTestSources]
:app:preBuild UP-TO-DATE
:app:preDebugBuild UP-TO-DATE
:app:checkDebugManifest
:app:preReleaseBuild UP-TO-DATE
:app:prepareComAndroidSupportAppcompatV72211Library UP-TO-DATE
:app:prepareComAndroidSupportSupportV42211Library UP-TO-DATE
:app:prepareDebugDependencies
:app:compileDebugAidl UP-TO-DATE
:app:compileDebugRenderscript UP-TO-DATE
:app:generateDebugBuildConfig UP-TO-DATE
:app:generateDebugAssets UP-TO-DATE
:app:mergeDebugAssets UP-TO-DATE
:app:generateDebugResValues UP-TO-DATE
```

```
Gradle build finished in 2s 776ms (3 minutes ago)
```

Fig. 5 Android Studio

4. CONCLUSION

This paper has shown what Android is and how it works with XML, JSON and API in combination with Java. It has elaborated how to create a development environment and the emulator (AVD). It has showed what Android Application Development [12] focus on, such as binding between XML and Java on Android. Different versioning of Android and its rapid evolvement in terms of new SDK's have been discussed as well as how developers take advantage of open source API`s. The paper has also coined the advantages with Android software environment, and has shown great environment for developer. Future of Android and its SDK is very wide. Android opens a wide area of possibility with new features and personal preferences, it has created a new era in sharing open source software equipment's, such free utilities and games, for mobile devices. Furthermore, Android has recently arrived, we believed that it will create the future because more than 50% of population is using and working on androidand the development environment and tools will be improved and enhanced in the future, making the development process accurate and effective.

5. REFERENCES

[1] What is android? http://developer.android.com/guide/basics/what-isandroid.html

[2] http://www.tutorialspoint.com/android/android_architecture.html

[3] http://www.slideshare.net/VijayRastogi/ppt2-intro-androidarchitecturecomponentsd6

[4] Static detection of malicious code in executable programs by J. Bergeron, M. Debbabi, J. Desharnais, M. M. Erhioui, Y. Lavoie, and N. Tawbi.

[5] Android Official Website (2008)—"Android | Official Website", <http://www.android.com/>.

[6] An Android Application Sandbox System for Suspicious Software Detection, by Thomas Bl¨asing, Leonid Batyuk, Aubrey-Derrick Schmidt, SeyitAhmetCamtepe, and Sahin Albayrak

[7] www.blackhat.com Jeff Lessard Gary C. Kessler, "Android Forensics: Simplifying Cell Phone Examinations" Small Scale Digital Device Forensics Journal Vol. 4, No.1, September 2010, Issn# 1941-6164,pp1-12.

[8] Tahat A.A. "Mobile personal electrocardiogram monitoring system and transmission using MMS"7th International Carbbean Conference on Devices, Circuits and Systems, 2008, 4:1-5.

[9] Feng Wang, Jiayu Zhu, MengShen, "A Communication Protocol of Wireless Sensor Network for Mobile Healthcare System" IEEE/ACM International Conference on Green Computing and Communications, pp 554-559, DOI 10.1109/GreenCom-CPSCom.2010.98.

[10] https://dspace.mah.se/bitstream/handle/2043/10721/AndroidApplicatcionDevelopment.pdf?sequence=1