

# Android Apps Development using Android Studio

N. Anju Latha<sup>\*</sup>, B. Rama Murthy

Department of Instrumentation, Sri Krishnadevaraya University, Anantapur, Andhra Pradesh India

## ABSTRACT

Android is a Linux-based open source software stack that comes along with operating system, middleware, native mobile applications, along with set of API libraries. It was designed primarily for touchscreen mobile devices such as smartphones, tablets. Android Studio is an integrated development environment (IDE) for Android platform development. Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. Android SDK provides set of application programming interfaces (APIs) and Android Studio Platform that can be used to create applications.

**Keywords:** Android, Android SDK, Android Studio.

## I. INTRODUCTION

Android is an open source and Linux-based Operating System for mobile devices such as smartphones 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 to develop only 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, whereas the first commercial version, Android 1.0, was released in September 2008.

In the present paper we explain necessary steps to develop Android based applications using the Android Studio Integrated Development Environment (IDE) and the Android 5 Software Development Kit (SDK). Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown in fig 1 of the architecture diagram.

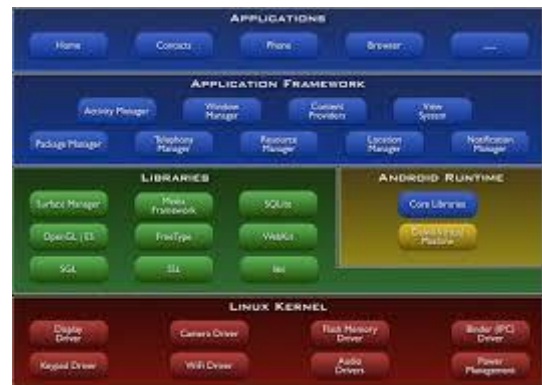


Figure 1: Android architecture diagram

### Linux kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

### Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit,

well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

### Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access. A summary of some key core Android libraries available to the Android developer is as follows

**android.app** : Provides access to the application model and is the cornerstone of all Android applications.

**android.content** : Facilitates content access, publishing and messaging between applications and application components.

**android.database** : Used to access data published by content providers and includes SQLite database management classes.

**android.opengl** : A Java interface to the OpenGL ES 3D graphics rendering API.

**android.os** : Provides applications with access to standard operating system services including messages, system services and inter-process communication.

**android.text** : Used to render and manipulate text on a device display.

**android.view** : The fundamental building blocks of application user interfaces.

**android.widget** : A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.

**android.webkit** : A set of classes intended to allow web-browsing capabilities to be built into applications.

Having covered the Java-based core libraries in the Android runtime, it is now time to turn our attention to the C/C++ based libraries contained in this layer of the Android software stack.

### Android Runtime

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called Dalvik Virtual Machine which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

### Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services –

Activity Manager : Controls all aspects of the application lifecycle and activity stack.

Content Providers : Allows applications to publish and share data with other applications.

Resource Manager : Provides access to non-code embedded resources such as strings, color settings and user interface layouts.

Notifications Manager : Allows applications to display alerts and notifications to the user.

View System : An extensible set of views used to create application user interfaces.

### Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

## II. METHODS AND MATERIAL

### Android Studio

Android Studio is an integrated development environment (IDE) for Android platform development. 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 JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

### Steps to create app in Android studio

We can start our application development by calling start a new android studio project. In a new installation frame should ask Application name, package information and location of the project as shown in fig 2.



Figure 2. new android studio project

After entered application name, it going to be called select the form factors your application runs on, here need to specify Minimum SDK, in our tutorial, I have declared as API21: Android 5.0(Lollipop).

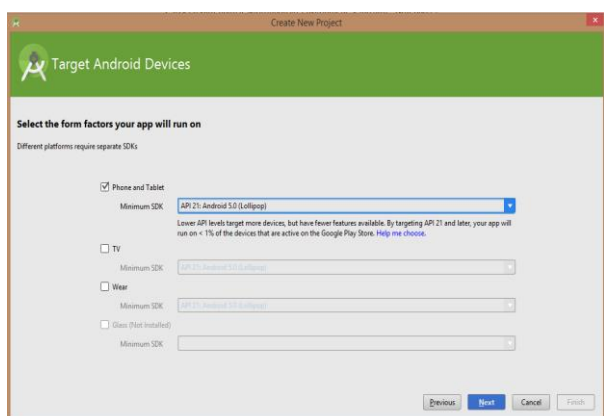


Figure 3 : Target Android devices selection

The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications

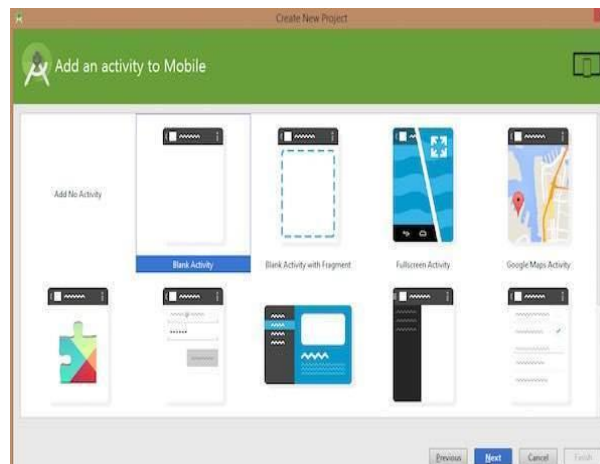


Figure 4 : Activity of Mobile

At the final stage it going to be open development tool to write the application code as shown in fig 5.

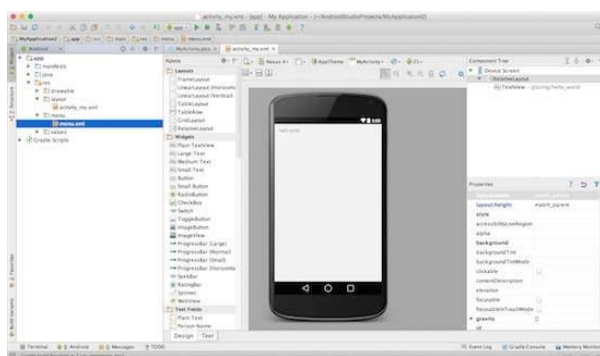


Figure 5 : Development Tool

### Create Android Virtual Device

To test our Android applications, we need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager Clicking AVD\_Manager icon as shown below



Figure 6 : Android AVD Manager

After Click on a virtual device icon, it going to be shown by default virtual devices which are present on

your SDK, or else need to create a virtual device by clicking Create new Virtual device button

If AVD is created successfully it means your environment is ready for Android application development. After we can close this window using top-right cross button or otherwise re-start the machine, now we are ready to proceed for our first Android example but before that we will see few more important concepts related to Android Application Development.

#### Hello Word Example

Before Writing a Hello word code, we must know about XML tags. To write hello word code, you should redirect to Activity\_main.xml file

To show hello word, we need to call text view with layout ( about text view and layout, you must take references at Relative Layout and Text View ).

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin"
  tools:context=".MainActivity">

  <TextView android:text="@string/hello_world"
    android:layout_width="550dp"
    android:layout_height="wrap_content" />
</RelativeLayout>
```

Need to run the program by clicking Run>Run App or else need to call shift+f10 key. Finally, result should be placed at Virtual devices as shown below fig.



**Figure 7 :** Virtual Device output

### III. CONCLUSION

The Android SDK carries a number of powerful tools to help with common Android development tasks. The Android emulator can be used for running and debugging Android applications virtually, without the need for an actual device

### IV. WEB REFERENCES

- [1] <http://developer.android.com/guide/basics/what-is-android.html>
- [2] [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [3] [http://www.tutorialspoint.com/android/android\\_architecture.html](http://www.tutorialspoint.com/android/android_architecture.html)
- [4] [http://dev.tutorialspoint.com/android/android\\_studio.html](http://dev.tutorialspoint.com/android/android_studio.html)