



Research on Development of Android Applications

Dr. D. M. Patel Prof. Dhruval Kachiya

- Sharma Sagar
- Gosai Dron
- Gaikwad Ritesh
- Baviskar Jay

Abstract

Gone are the days when the mobile phone had to ring to capture our attention or the computer was the only device people used. The mobile application field has been rising at a tremendous rate with the drastic increase in the number of mobile apps in various mobile phones and tablets. Mobile apps are essentials they provide functionalities that can server useful purposes such as finding a location or booking movie tickets online. In today's fast paced world, mobile marketing is becoming very competitive. To ensure visibility of your app in such complex scenario, a specific approach needs to be followed to ensure a successful app development. In this paper, numerous factors that can play a significant role in successful app development are discussed with specific examples and explanation.

I. INTRODUCTION

Mobile Application Development refers to the process of making application software for handheld devices such as mobile phones and Personal Digital Assistants. Through the usage of mobile apps, the user is provided with various features that will enable him to full-fill all his needs and much more. Apps should be interactive to the users. Apps can be downloaded from various platforms such as Google Play Store and iOS App Store. There are free apps as well as paid apps. Some apps can be used for free for a specific amount of time before subscribing for premium membership.

For developing apps, the constraints and features of mobile devices needs to be considered. For example, mobile devices have lower processing power, run on battery but have more features such as location detection. Wide range of screen sizes and hardware specifications also needs consideration. For developing apps, specialised integrated development environments such as **Android Studio** or **Eclipse** is required. The app is first tested using devices called emulators which is a software simulation of the actual hardware device and then finally field testing is performed. Mobile user interface(UI) Design is

another essential part in application development. The UI involves considerations of contexts, screen and user input and output mobility. The user manipulates the application via input and then the expected results are displayed via the output. There are mobile UI constraints such as limited screen size. Mobile UI is considered as front-end and they rely on back-ends to support access to enterprise systems. The back-end facilities include data routing, security and authorisation and is provided by middleware components such as Mobile Backend as a service (MBaaS).

II. FEATURE OF ANDROID

Sr.No	Feature & Description
1	<p>Beautiful UI</p> <p>Android OS basic screen provides a beautiful and intuitive user interface.</p>
2	<p>Connectivity</p> <p>GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.</p>
3	<p>Storage</p> <p>SQLite, a lightweight relational database, is used for data storage purposes.</p>
4	<p>Media support</p> <p>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.</p>
5	<p>Messaging</p> <p>SMS and MMS</p>
6	<p>Web browser</p> <p>Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.</p>
7	<p>Multi-touch</p> <p>Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.</p>
8	<p>Multi-tasking</p> <p>User can jump from one task to another and same time various application can run simultaneously.</p>
9	<p>Resizable widgets</p> <p>Widgets are resizable, so users can expand them to show more content or shrink them to save space.</p>
10	<p>Multi-Language</p> <p>Supports single direction and bi-directional text.</p>

III. ANDROID ARCHITECTURE

- Android is an open source, Linux-based software stack created for a wide array of devices and form factors.
- Android architecture contains different number of components to support any android device needs.
- Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.
- The main components of android architecture are following:-
 - Applications
 - Applications Framework
 - Android Runtimes
 - Android Libraries
 - Linux Kernel



➤ APPLICATION –

- Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third party applications downloaded from the play store like chat applications, games etc.

- It runs within the Android run time with the help of the classes and services provided by the application framework.
- Android applications are written in the Java-programming language. The java code is assembled with the data related to application and resource files. All these is bundled by the app tool into an android package. It is an archived file with suffix of apk. This file works as a lorry for issuing the application which is installed on mobile devices.
- By default, every application has its own Linux course. Android run the application when it is desired by the user and close the application when it is no longer essential. As the assets are required by the system.
- Each application code run separately. Each process has its own JVM Java virtual machine.

➤ APPLICATION FRAMEWORK –

- Application Framework provides several important classes which are used to create an Android application.
- It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.
- It includes different types of services activity manager, notification manager, view system, package manager etc. which are helpful for the development of our application according to the prerequisite.

➤ APPLICATION RUNTIME –

- Android Runtime environment is one of the most important part of Android.
- It contains components like core libraries and the Dalvik virtual machine(DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries.
- Like Java Virtual Machine (JVM), **Dalvik Virtual Machine (DVM)** is a register-based virtual machine and specially designed and an optimized for android to ensure that a device can run multiple instances efficiently.

➤ ANDROID LIBRARIES –

- The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.
- Bionic system C library: it is the C language standard library, and also the bottom library of the system, which is invoked by Linux system.
- Media Framework: based on the PacketVideo, OpenCORE to support the player back and recording of audio and video in multiple formats, such as MPEG4,MP3, AAC, AMR, JPG, PNG.
- SGL[scalable graphics library] : 2D graphics engine library.
- SSL[secure socket layer] : located between TCP/IP protocol and other application protocols to support data communication.
- OpenGL ES1.0: support for 3D effect.
- SQLite: relational database.

- Webkit: Web browser engine.

➤ LINUX KERNEL –

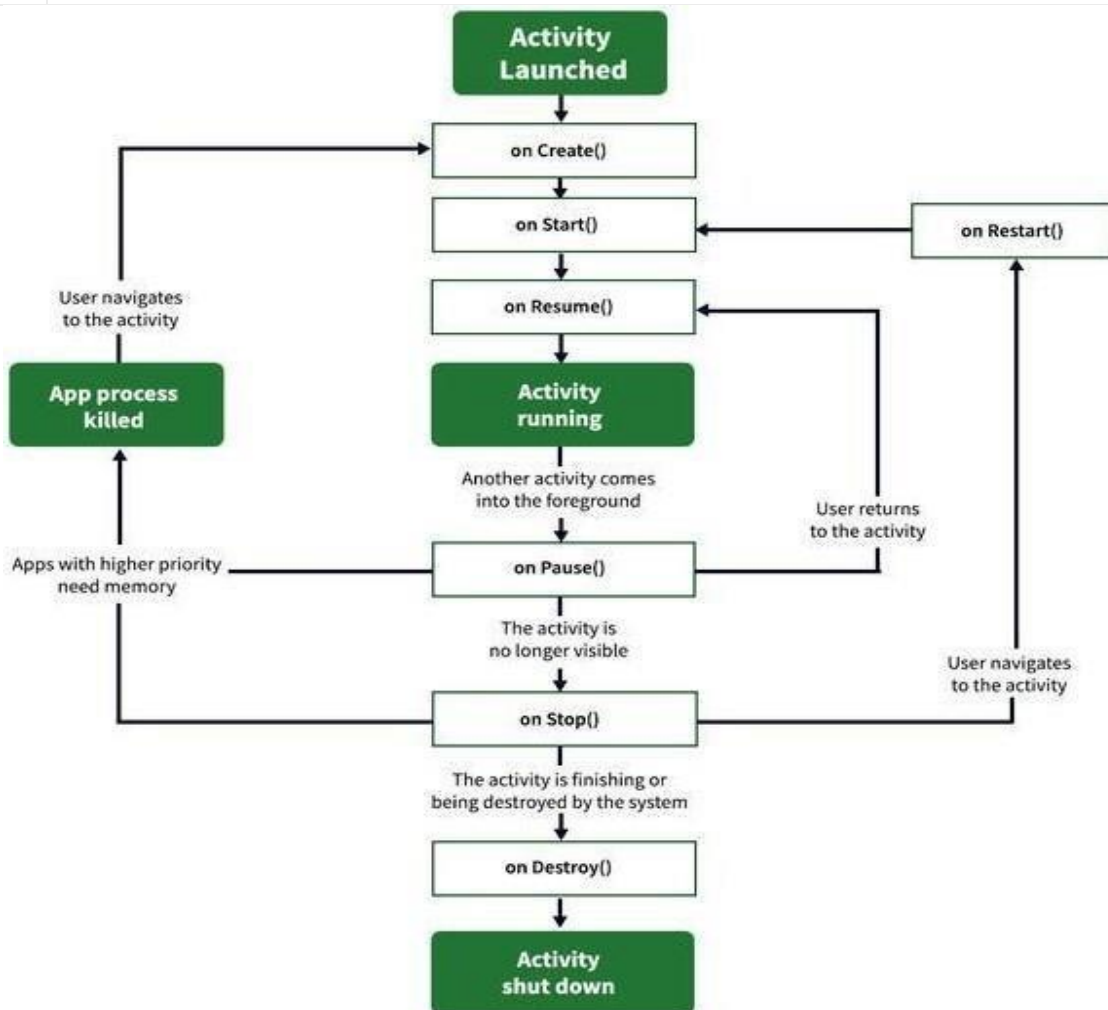
- Linux Kernel is heart of the android architecture.
- It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc.
- It is responsible for management of memory, power, devices etc.

IV. ACTIVITY LIFECYCLE IN ANDROID

- In Android, an activity is referred to as one screen in an application. It is very similar to a single window of any desktop application. An Android app consists of one or more screens or activities.
- Each activity goes through various stages or a lifecycle and is managed by activity stacks.
- An activity is the single screen in android. It is like window or frame of Java.
- Activity is one of the building blocks of Android OS. In simple words Activity is a screen that user interact with.
- Every Activity in android has lifecycle like created, started, resumed, paused, stopped or destroyed.
- There are Seven stages of an activity:-

Sr.No	Callback & Description
1	onCreate() This is the first callback and called when the activity is first created.
2	onStart() This callback is called when the activity becomes visible to the user.
3	onResume() This is called when the user starts interacting with the application.
4	onPause() The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed.

5	onStop() This callback is called when the activity is no longer visible.
6	onDestroy() This callback is called before the activity is destroyed by the system.
7	onRestart() This callback is called when the activity restarts after stopping it.



Activity Lifecycle in Android

1. `onCreate()`
2. `onStart()`
3. `onRestart()`
4. `onResume()`
5. `onPause()`
6. `onStop()`
7. `onDestroy()`

V. NEW FEATURES OF ANDROID APPLICATIONS

As a young operating system, on one hand, Android could benefit from mature technology of other operating system. On the other hand, Android could also improve the blemish that appears in other operating systems. On a developer's prospective, Android possess following new features:

The permission that an application possessed has been defined clearly. In android applications, all components that could be independently launched by system need to be declared in a XML file named AndroidManifest. The AndroidManifest does a number of things in addition to declaring the application's components, including:

- Identify any user permissions the application requires, such as Internet access. Only identify the permissions that an application requires, the application have permissions to perform operations.
- Declare the minimum API level required by the application.
- Declare hardware and software features used or required by the application.
- Declare API libraries the application needs to be linked against.

Resources are separate from the source code. In Android, all non code resources are defined in XML files. For every resource that included in an Android project, the SDK build tools define a unique integer ID, which can be used to reference the resource from the application code or from other resources defined in XML files. Providing resources separate from source code makes it easy to update various characteristics of an application without modifying code and by providing sets of alternative resources enables developers to optimize the application for a variety of device configurations, such as different languages and screen sizes.

VI. CONCLUSION

Android as a full, open and free mobile device platform, with its powerful function and good user experience rapidly developed into the most popular mobile operating system. This article gives a detailed introduction of Android application framework and the working principal of Android applications. Finally, a music player on the android platform was put forward as an example to illustrate this mechanism.

VII. REFERENCES

- [1]. Ribeiro and A. R. D. Silva, "Survey on Cross-Platforms and Languages for Mobile Apps," Eighth International Conference on the Quality of Information and Communications Technology, 2012.
- [2]. Pohares, V. C. Kulloli, T. Bhattacharyya, and S. Bhure, "Cross Platform Mobile Application Development," International Journal of Computer Trends and Technology, vol. 4, no. 5, pp. 1095–1100, 2013.
- [3]. S. S. Jagtap and D. B. Hanchate, "Development of Android Based Mobile App for PrestaShop eCommerce Shopping Cart (ALC)," International Research Journal of Engineering and Technology
- [4] Micro computer information, 2011,27 (09): 1-3. Zeng Jianping, Shao Yanjie. Research on Android system architecture and application development [J].
- [5] Yao Yumin, Liu Weiguo. Research on Android architecture and application development [J]. Computer system application, 2008 (11): 110-112 + 24.