

Android App Development: A Review

Anmol Tewari¹, Pawan Singh²

Amity School of Engineering and technology, Amity University, Lucknow, India^{1,2}
anmoltewari15@gmail.com¹, pawansingh51279@gmail.com²

How to cite this paper: A. Tiwari and P. Singh (2021) Android App Development: A Review. *Journal of Management and Service Science*, 1(2), 6, 1-6.

<https://doi.org/10.54060/JMSS/001.02.006>

Received: 24/02/2021

Accepted: 07/03/2021

Published: 08/08/2021

Copyright © 2021 The Author(s).

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Android is a software stack for mobile devices that includes an operating system, middleware, and key applications. Android is a software platform and operating system for mobile devices based on the Linux operating system and developed by Google and the Open Handset Alliance. It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries but does not support programs developed in native code. The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008, most of the Android platform will be made available under the Apache free-software and open-source license.

Keywords

Android Studio, components, Android Libraries, Architecture, UI Design

1. Introduction

Mobile technology is the expertise castoff for cellular messaging. Subsequently the start of the generation of mobile to present, a typical smart phone maneuver has left from being a modest shared pager to being a smart device, Global Positioning System navigation device, a surrounded web browser in addition instantaneous messaging user, and a hand-held gaming control box. Many of portable functional structures are available for smart receivers which include Windows Mobile, Android, iOS, BlackBerry OS, Bada. Android is an operating system based on Linux with a Java programming interface. Android is a portable operational system (OS) developed by Google. Android is the primary completely open-source mobile OS. All the android versions are in the designation of dissimilar chocolates e.g. are Cupcake, Donut, Éclair, Froyo, Gingerbread, HoneyComb, Ice-cream Sandwich, Kitkat, Lollipop, Marshmallow, Nougat, Pie. A mobile app is a computer software tender premeditated to course on mobile, tablet computers and different other mobile devices. Smart phone apps remained initially accessible for wide-ranging yield in addition data repossession, which includes electronic mail, datebook, associates, money market in addition meteorological conditions information.

2. Technology Used

Android Studio is used to implement and code the applications.

2.1. Android Architecture

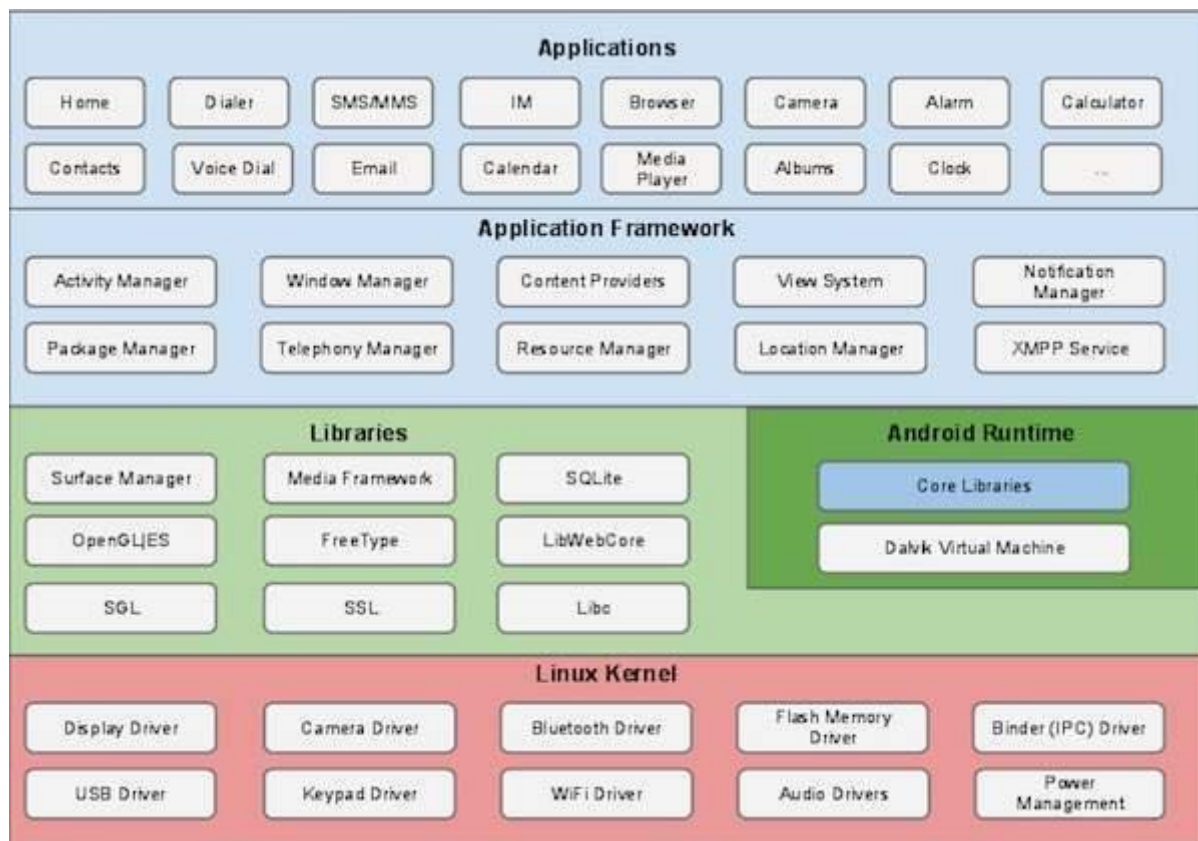


Fig. 1. Android Architecture

2.2. Hardware

Google has unveiled at least three prototypes for Android, at the Mobile World Congress on February 12, 2008. One prototype at the ARM booth displayed several basic Google applications. A 'd-pad' control zooming of items in the dock with a relatively quick response. A prototype at the Google IO conference on May 28, 2008 had a 528 MHz Qualcomm processor and a Synaptic capacitive touch screen and used the UMTS cellular standard. It had 128 MB of RAM and 256 MB of flash, showing that Android's memory requirements are reasonable. The demo was carried out using a .6 Mbps HSDPA connection.

3. Building Block

3.1. Activity

It is a tender module which delivers a display using which operators could cooperate in command to perform approximately, like business the itinerant, snap a picture, guide the email, or view a record. Individually movement is prearranged a window in which to draw its manipulator boundary. The window naturally plugs the shelter. Whenever an original activity

starts, it is pushed onto the vertebral heap and takes user focus. The back stack abides to the basic “last in, first out” stack mechanism, so, when the user is done with the current activity and presses the Back button, it is popped from the stack (and destroyed) additionally the preceding action resumes. When an activity is stopped because an innovative activity starts, it is notified of this change in state through the activity’s lifecycle callback methods.

3.2. Services

A package is a tender module which could accomplish extensive successively performing processes on the circumstantial, in addition it prepares not to offer a operator boundary. Additional tender module could jump a service, and it continues to course in the circumstantial level though the operator shifts to different tenders. There are quite a few recall methods that an activity might receive, owing to an adjustment in its state whether the system is creating it, stopping it, resuming it, or destroying it- and each callback provides you the opportunity to perform specific work that is appropriate to that state change. For instance, when stopped, our activity should release any large objects, such as network or database connections.

3.3. Content Providers

A data worker achieves a collective usual of tender information. We could collect the information in the file system, an SQL information record, scheduled the network, or somewhat additional tenacious storage location your tender could entrée. A data worker delivers a structured interface to application data. Through a data worker our tender could segment information using some additional requests. Android contains an SQLite database that is customarily used in unification with a content provider. When the activity resumes, we can reacquire the necessary resources and resume actions that were interrupted. These state transitions are all part of the activity lifecycle.

3.4. Broadcast Receivers

Pixabay is an international, free to use website used for sharing pictures, vector images, illustrations, video clips as well as audio clips. Currently Pixabay offers approximately 1188454 photos, vector graphic images, videos, and audio clips free of cost. All the image and video files available on the website can be utilized for both personal and commercial purposes without any attribution requirement. Transmission Telephones merely retort to transmission communications from further tenders or beginning the arrangement the aforementioned. For example, tenders can similarly recruit programs to let additional tenders see that roughly information has been transferred to the phone and is obtainable for them for usage, hence it is transmission phone which will seize this announcement in addition will inductee suitable achievement.

4. Android Libraries

This class incorporates that Java-built in functions which were explicit to Android advancement. Some of the examples of built-in functions in cutting-edge sort comprise of the fond summary functions additionally of that whose simplify handler boundary edifice, illustrations sketch besides record entrée. An immediate off around crucial central Android built in functions which are accessible to the originator is as per the following: -

- android.app- Propositions installment to tender prototype additionally is the groundwork of all Android applications.
- android.content- Simplifies data entrée, issuing also communication among tenders and tender gears.
- android.database- Castoff to entrée information available by pleased workers and comprises SQLite information record organization modules.
- android.opengl- A class to the OpenCV EA 3D illustrations translation application programming interface.



- android.os- Delivers requests by entrée to typical operating system amenities which comprises posts, scheme facilities then intra-procedure announcement.
- android.text- castoff to essence on addition to deploying typescript on an exercise monitor
- android.view- the fundamental building block of application user interfaces
- android.widget- an ironic assembly of initial made operator boundary workings such as buttons, labels, listviews, outline executives, radio switches and so on.

4.1. Authors and Affiliations

Menu component is responsible for generating the navigation bar of this website. It uses standard bootstrap navigation with some custom css styling to achieve so. menu component once loaded, stays there throughout the life cycle of this website. Property Binding While developing Angular applications, developers come across a couple of methods for displaying data on the view. First is property binding and the other is interpolation. They are used to transfer data from component to template. Property binding can be defined as a one-way technique that allows to set the attribute of the view element. Property binding involves updating the property value present in component and then bind it with an element present in the view template. Attributes which are bound to an element are always placed within square brackets. They are uno of the simplest layout types cast-off to establish controls within the user interface of an Android application. They will be worked fewer frequently than some other layouts, simply since they are normally castoff to display only one view, or views which overlap. The efficiency of a frame outline makes it a upright optimal for screens containing few view controls (home screens, game screens with a single canvas).

Sometimes other inefficient outlines strategies can be reduced to a frame layout design that is extra effective, while other times a more specialized layout type is appropriate. Frame layouts are the normal layout of choice when you want to overlap views.

4.2. Identify the Headings

They are single of the meekest and utmost mutual categories of strategies used by Android developers to organize controls within their user interfaces. The linear layout works much its name implies: it organizes controls linearly in either a vertical or horizontal fashion. When the layout's orientation is set to vertical, all child controls within it are organized in a single column. when the layout's orientation is set to horizontal, all child controls within it are organized in a single row.

Some of the most important attributes you will use with linear layouts include:

- The orientation attribute (required), which can be set to vertical or horizontal.
- The gravity attribute(optional), which controls how all child controls are aligned and displayed within the linear layout (class: LinearLayout)
- The layout weight attribute (optional, applied to each child control) specifies each child control's relative importance within the parent linear layout (class: Linear Layout. Layout Params Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

4.3. Relative Layout

The relative layout works much as its name implies: it organizes controls relative to one added, or onto parent control itself. It resources that kid controls, such as ImageView, TextView, and Button controls, can be placed above, below, to the left or right, of one additional. Teen-ager joysticks can also be placed in relation to the paternal (the relative layout container); including

placement of controls aligned to the top, bottom, left or right edges of the layout. Some specific attributes apply to relative layouts- namely the child rules, including:

- Rules for child control centering within the parent layout, including: center horizontally, center vertically or both.
- Rules for child control alignment within the parent layout, including: align with top, bottom, left or right edge of another control.
- Rules for child control alignment in relation to other child controls, including: align with top, bottom, left or right edge.
- Rules for child control placement in relation to other child controls, including: settlement to the leftward or right of a detailed control, or above or below another control.

4.4. Table Layout

A table layout is exactly what you might expect: a grid of made up of rows and columns, where a cell can display a view control. From a user interface design perspective, a `TableLayout` is comprised of `TableRow` controls_ one for each row in your table. The contents of a `TableRow` are simply the view controls that will go in each “cell” of the table grid. Although table layouts can be used to design entire user interfaces, they usually aren’t the superlative device for undertaking so, as it has been consequential from `Linear Layout`.

4.5. Options Menu and Action Bar

The options menu is the primary collection of menu items for an activity. It’s where you should place actions that have a global impact on the app, such as “Search”, “compose email”, and “settings”. On Android 3.0 and higher, items from the options menu are accessible by the achievement bar as an amalgamation of on-screen action items and overflow options. Beginning with Android 3.0, the `Menu` button is deprecated (some devices don’t have one), so we should migrate toward using the action bar to provide access to actions and other options.

4.6. Context Menu and Contextual Action Mode

A context menu is a floating tariff that look as if as soon as the manipulator performs a long click on an element. It provides travels that mark the designated content or context frame. A contextual tariff bids schedules that mark the detailed entry or situation frame in the UI. You can provide a context menu for any view, nevertheless they are maximum repeatedly used for items in a `ListView`, `GridView`, or other view collections in which the manipulator can perform direct actions on each item.

4.7. Popup Menu

After building all those components, we need to somehow relate them to enable navigation between them. Some of these events driven navigation is directly controlled in respective typescript files while the static ones are handled using routing module. For eg: A invalid path results in 404 error, this is handled by redirecting any such requests to home component.

4.8. Adapters

This is a custom service designed to handle authentication with firebase auth, thereby simplifying this process and reducing repetition of code. Making a custom service enables us to change authentication method in future, without struggling too much as now all the authentication code is at one place and not scattered among several files.

4.8. Notification Manger

This is a custom guard designed to validate email address, password etc. before it is pushed to database. Similar to auth service, we define it as a guard to reduce repetition of code and gives us flexibility to tune these criteria with a lot more ease. This guard is used in login, signup, and profile component to validate the information given by the user. Guards are implemented to impose constraints on information provided by user, such as password, email id, name etc.

Conclusion

Android smartphones are in the hype in the 21st century. The scope of android applications is increasing day by day. Its development has become an essential part of today's programming curriculum. Its utility and efficiency are also high. The society has no dearth of ideas. These ideas can be most effectively implemented by developing user-friendly android applications. Android is a much more diverse operating system than iOS and Windows Phone Mobile. Android has grown rapidly over the past 4 years becoming the most used smartphone operating system in the world. It is because Android doesn't release 1 phone from 1 company with 1 new OS every year, but countless phones from numerous companies, adding their own twist, throughout the year, developing gradually day-by-day. Android's ability to customize is unparalleled compared to Apple's and Microsoft's software allowing the user to change and customize nearly every aspect of Android which most iPhone and Windows users would not dream. So, Android is unique and incomparable to other mobile operating systems.

Acknowledgements

I would like to thank my university for giving me this golden opportunity to research on such an interesting topic. I would also like to thank Dr. Anil Kumar, Assistant Pro Vice Chancellor & Director, ASET and Dr. Deepak Arora, Professor & Head, Dept of CSE & IT, ASET for encouraging students to indulge in valuable research activities to enhance their technical skills. Next, I want to thank my faculty guide Dr. Pawan Singh for guiding me throughout the course of this project. I would also like to thank the internet, books, and the institute for the knowledge I acquired with their help.

References

- [1] A. Kathuria and A. Gupta, "Challenges in android application development", International Journal of Computer Science and Mobile Computing, vol.4, no.5, pp 294-299, 2015.
- [2] C.K. Marigowda, "Android based college application for students," Int. j. innov. res. adv. eng., vol. 8, no. 1, pp. 7–14, 2021.
- [3] P. Sankhe, R. Srivastava, and V. Prasad, "Android Application for College Management System" International Journal for Scientific Research and Development, vol.6, no.12, pp. 3699-3701, 2016.
- [4] S. Doğan, A. B. Can, and V. Garousi, "Web application testing: A systematic literature review," J. Syst. Softw., vol. 91, pp. 174–201, 2014.
- [5] L. Cruz and R. Abreu, "Performance-based guidelines for energy efficient mobile applications," in IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft), 2017.
- [6] A. Ribeiro and A. R. da Silva, "Survey on cross-platforms and languages for mobile apps," in Eighth International Conference on the Quality of Information and Communications Technology, 2012.
- [7] F. Appiah, J. B., J. K., et al, "A tool selection framework for cross platform mobile app development," Int. J. Comput. Appl., vol. 123, no. 2, pp. 14–19, 2015.
- [8] M. Brambilla, A. Mauri, M. Franzago, et al, "A model-based method for seamless web and mobile experience," in Proceedings of the 1st International Workshop on Mobile Development, 2016.
- [9] A. Kathuria and Anu Gupta, "Challenges in android application development" International Journal of Computer Science and Mobile Computing, vol.4, no.5, pp 294-299 May 2015.
- [10] V. Deshmane, S.Sawale 2 , K. Bharambe ,et al, "Application Development with Android: A review" (IJCESR), vol.5, no.4, pp. 2394-0697, 2018.

