INTRODUCTION

Real-time android quiz app titled “QuizUp” is developed for quiz competition with all other online users. QuizUp is a multi-player game in which one user competes against another during seven rounds of timed multiple-choice questions of various topics. User have the ability to either challenge a friend to a match or be paired against a random player. Users have the option of signing into the app through a social media platform Google+.

Backend: -

To perform real-time competition, the app uses Google’s Firebase cloud messaging services. Firebase gives the tools to develop high-quality apps. Firebase Real-time Database store and sync with NoSQL cloud database. Data is sync all clients on real-time, and remains available when app goes offline. The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in real-time to every connected client. All of the clients share one Real-time Database instance and automatically receive updates with the newest data. All messages of this app are store on Firebase database to achieve Real-time service.

Frontend: -

This app is design to run on any Android devices having different screen resolution like Android phone, Android T.V., Android TAB etc. Scroll layout can dynamically handle different sizes of screens. Different color combinations make the app to look more realistic. Text sizes, text styles and color can be customized by user. The app uses XML (Extensible Markup Language) for front-end. XML is like HTML, it’s a tag language used to design realistic layout. The app uses Java at the back end of the app.

Development

Development of QuizUp Android app is hard work for a beginner developer. It required advance knowledge of Java programming language and good work experience on Android Studio - Integrated Development Environment. For working with real time database, a good knowledge is also required on Firebase Web Services. Some additional knowledge is required on SQLite database, Google+ login integration, Android animations & material designing.

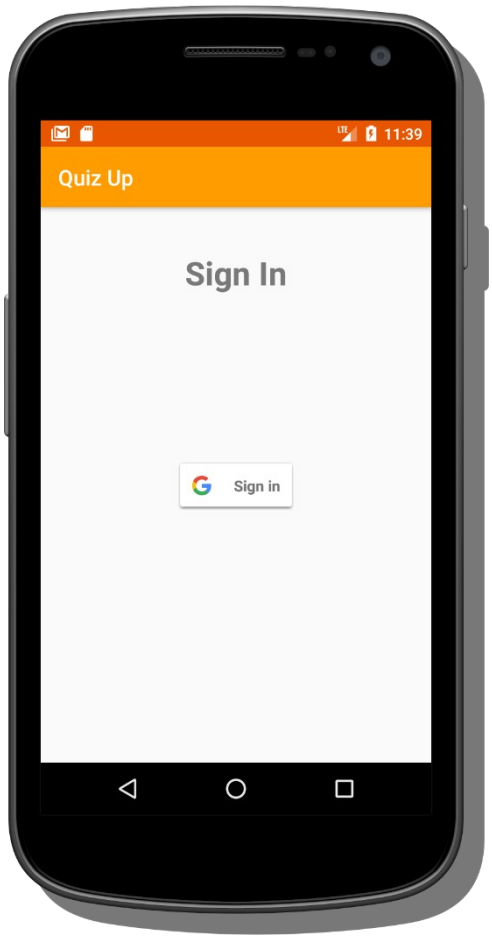
- **Splash Screen**
  Android splash screen is used so that apps can show their brand icons before showing the content of app. This may also use to do some background work in the application like loading resources from network while the splash screen is being shown. This will look smooth for user. In QuizUp app the splash screen is doing the same work and also a button is use to change the activity.

- **Topic Activity**
  User need to choose a topic in various options. According to the topic selected, the app going to search online users. If user select a Java topic for real time quiz, all online opponents can challenge him for only Java. There are many topics, user can play a quiz on only one topic at a time.

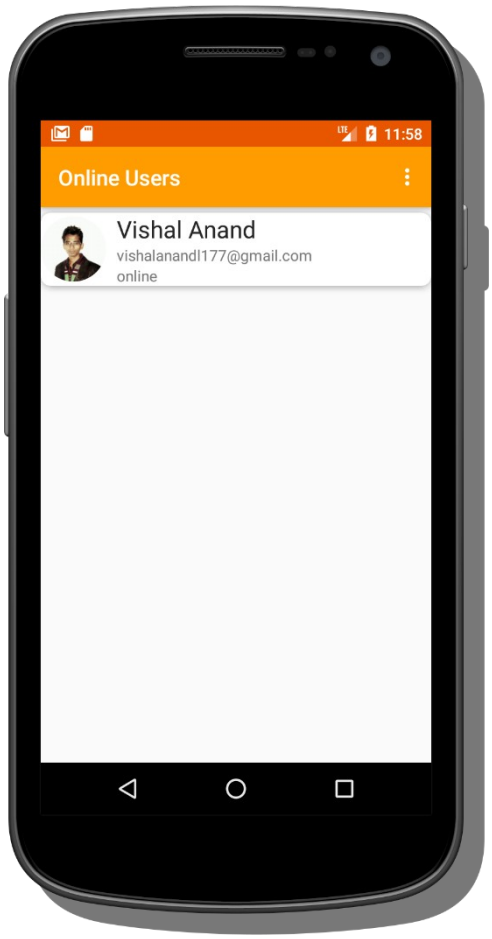
- **Sign in Activity**

  User is required to sign in before starting the quiz competition. User can sign in using there Google+ account just simply by clicking the sign in button, this leads you to the “choose account for QuizUp” prompt. By clicking the appropriate account user will able sign in into the app. No further information is required.

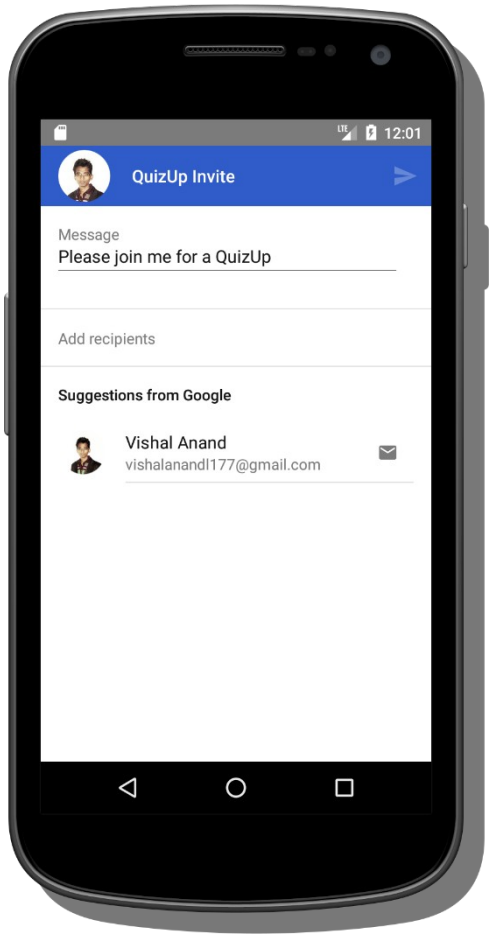
  Signing in is required to store user’s score and other information. This app only store user’s name, email, profile photo URL, and UID generated by Google itself.



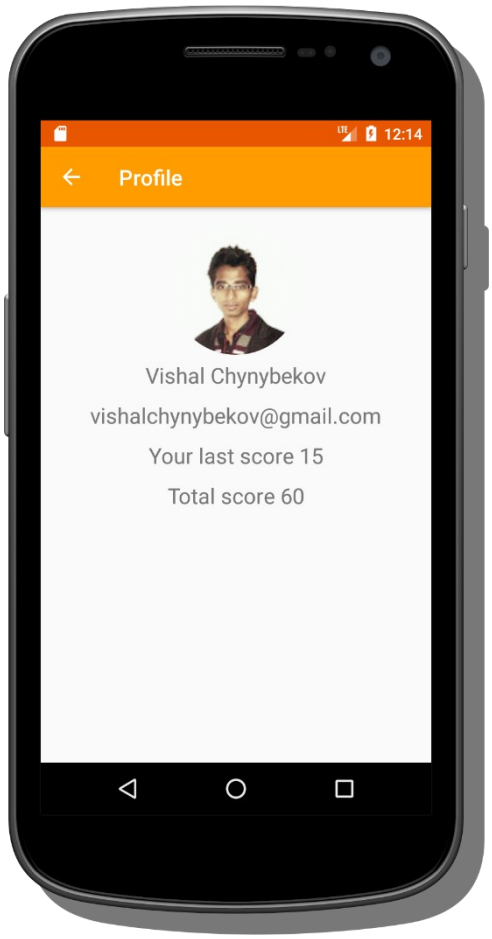
- **Online Users Activity**
  By choosing appropriate topic, you can choose any opponent from a list of online users. This list of online users is real time, so any user goes offline or come online, the list will reflect immediately.



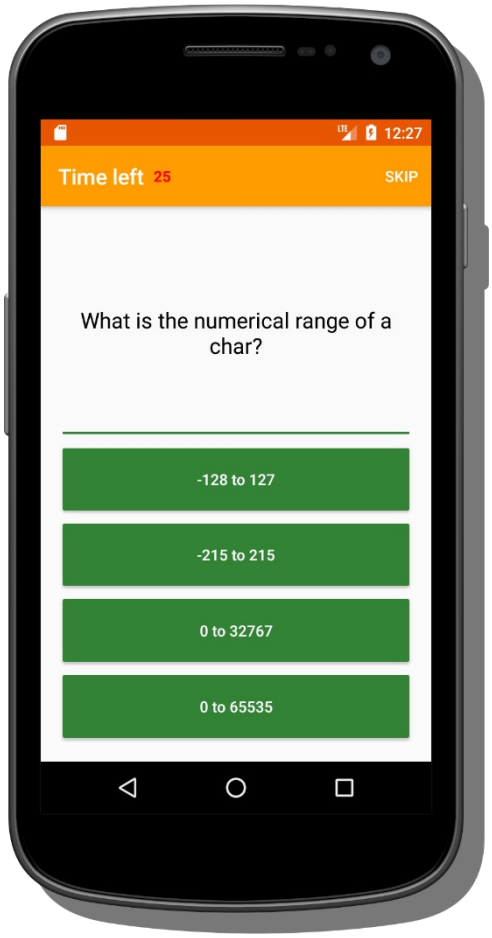
- **Invite Friend Activity**
  User can invite friends or family to play quiz with him. Using friends email, user can send email to join him for QuizUp. User can even send a message on a friend's contact number just by typing the contact number or by choosing the contact from the phone book. If friend have this application already installed, then by clicking the link in the email or message app will automatically open.



- **Profile Activity**
  In this activity user can see their last score and total score.



- **Quiz Activity**
  Quiz Activity consists of adding of multiple questions on the server side realtime database so the app can download the question within a second, a request option so the user can request to other remote side opponent user to accept the challenge, a confirmation option to the opponent user to accept the challenge, a threaded series of question handling on the client side so the app can wait for a particular time to accept the user input and when the time is complete then the app can ask another question.
  A user has only 30 seconds to answer display on the top of the screen (Action bar). User can skip the question by pressing the skip button.
  This activity has a question section and four options. By choosing the option, if user is wrong then he will get no score or points, if he is right the for every right answer he will get +5 score or points. Answering right makes user's profile more strong.



Android

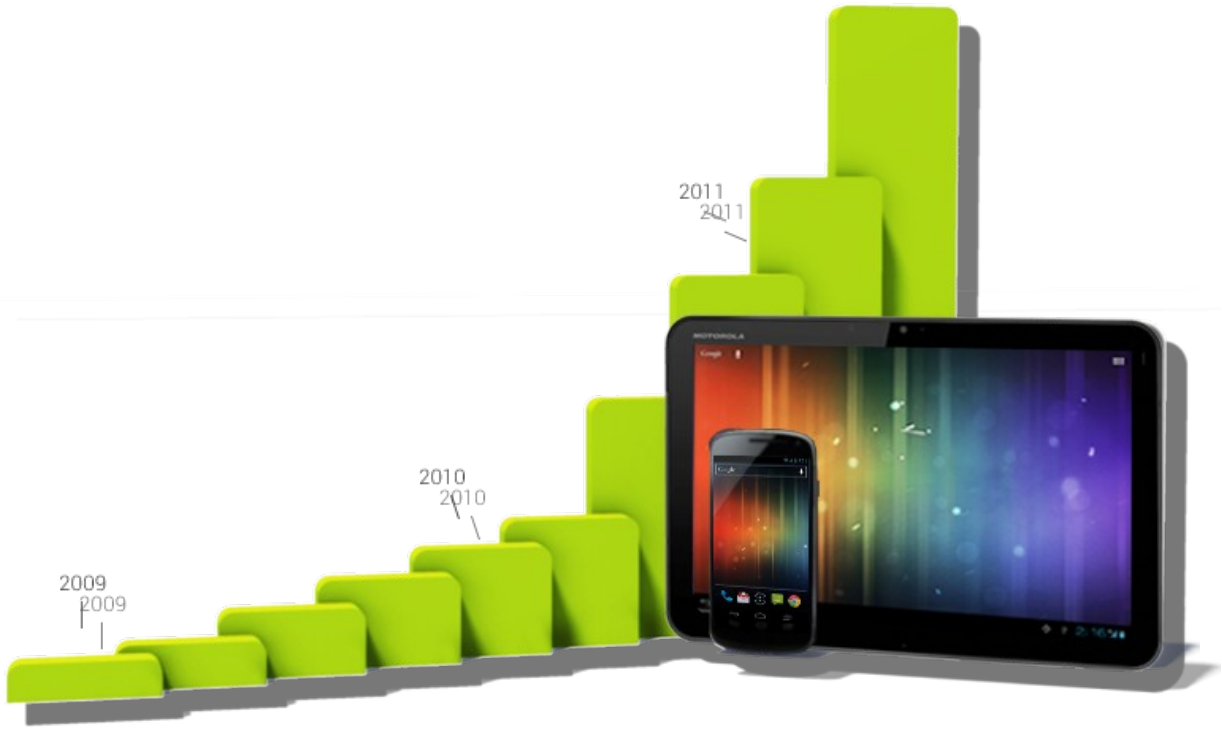
Android, the world's most popular mobile platform.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast—every day another million users power up their Android devices for the first time and start looking for apps, games, and other digital content.

Android gives you a world-class platform for creating apps and games for Android users everywhere, as well as an open marketplace for distributing to them instantly.

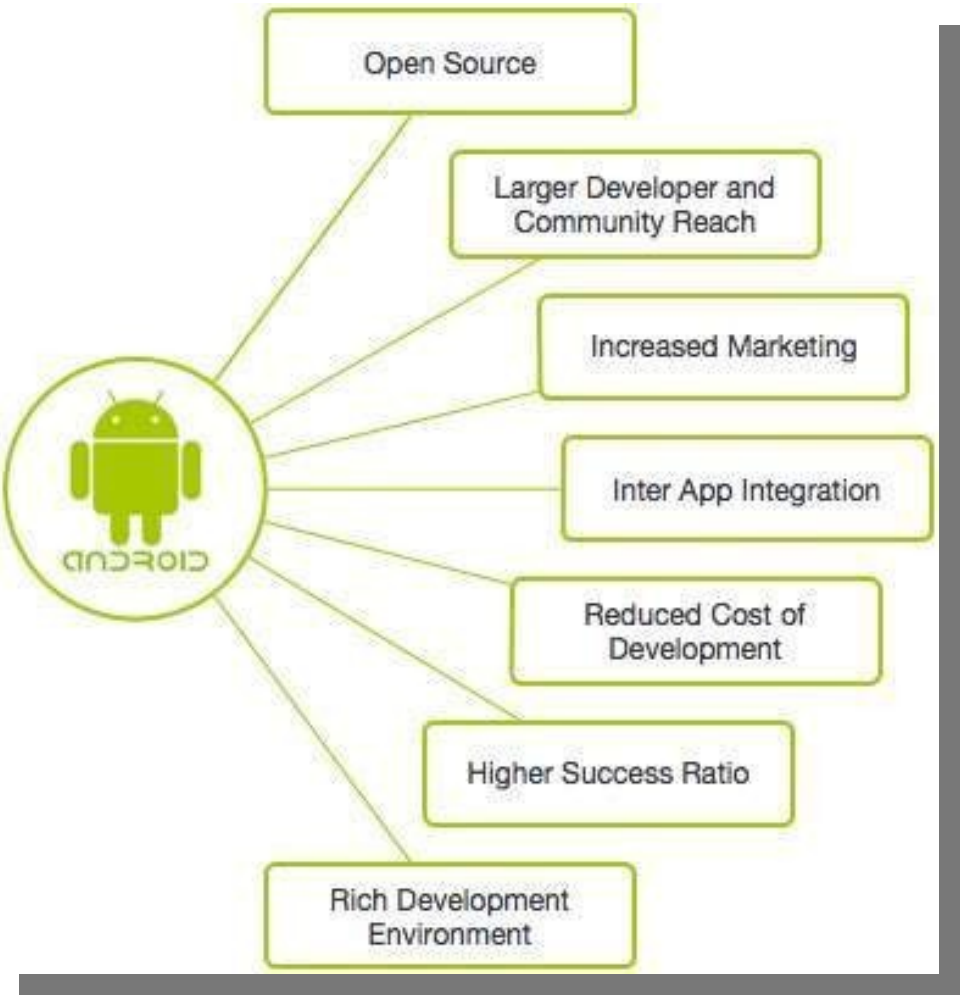
Android is an open source and Linux-based operating system for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.



Why Android ?

Android is most powerful development framework. Google Play is the premier marketplace for selling and distributing Android apps. When you publish an app on Google Play, you reach the huge installed base of Android.



Features of Android

- Beautiful UI

  Android OS basic screen provides a beautiful and intuitive user interface.

- Connectivity

  GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.

- Storage

  SQLite, a lightweight relational database, is used for data storage purposes.

- Media support

  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.

- Messaging

  SMS and MMS

- Web browser

Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.

- Multi-touch

  Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.

- Multi-tasking

  User can jump from one task to another and same time various application can run simultaneously.

- Resizable widgets

  Widgets are resizable, so users can expand them to show more content or shrink them to save space.

- Multi-Language

  Supports single direction and bi-directional text.

- GCM

  Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.

- Wi-Fi Direct

  A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.

- Android Beam

  A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

Components & Description

- Activities

  They dictate the UI and handle the user interaction to the smart phone screen.

- Services

  They handle background processing associated with an application.

- Broadcast Receivers

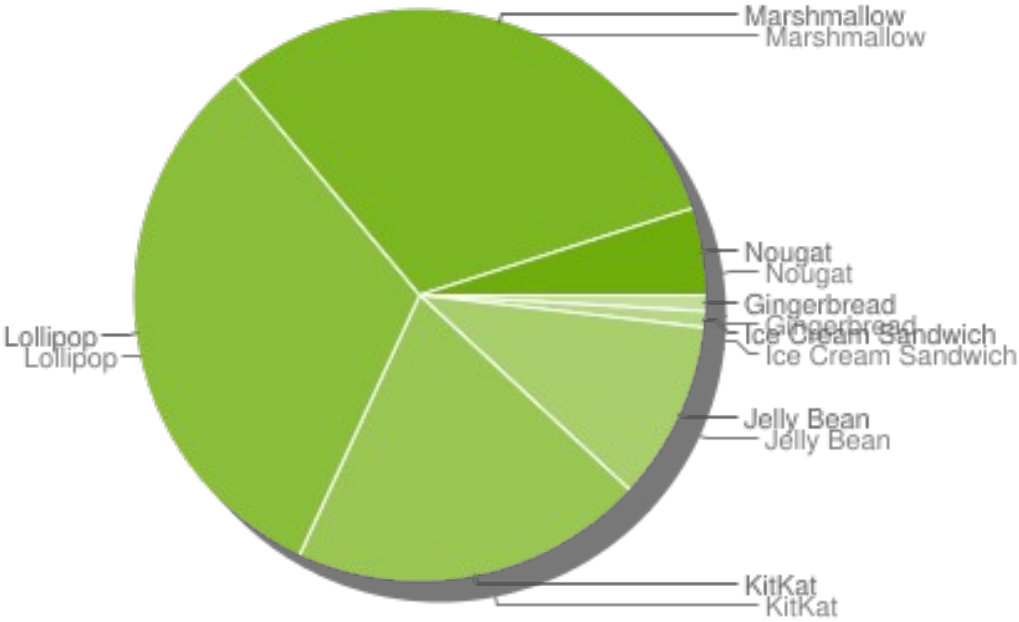
  They handle communication between Android OS and applications.

- Content Providers

  They handle data and database management issues.

Platform Versions

Data collected during a 7-day period ending on April 3, 2017.



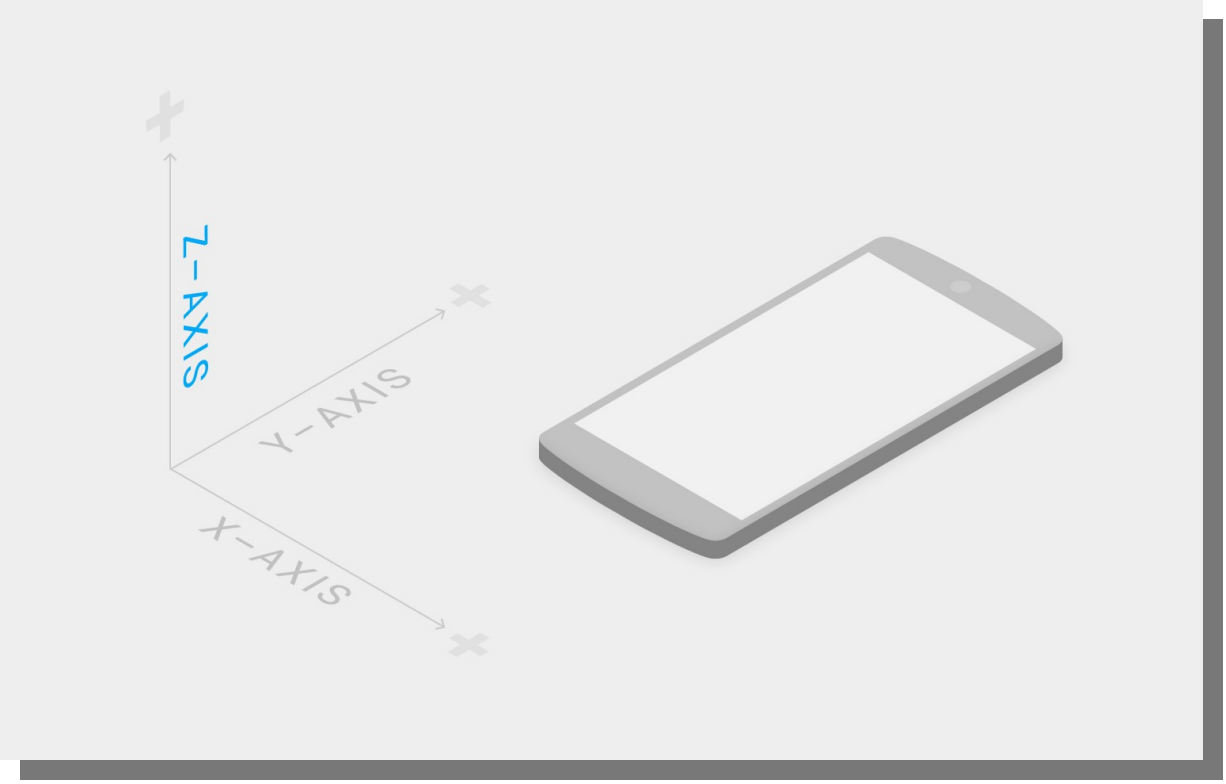
Android Material Designing

Android uses a new design metaphor inspired by paper and ink that provides a reassuring sense of tactility.

Material design is a three-dimensional environment containing light, material, and cast shadows.

The material environment is a 3D space, which means all objects have x, y, and z dimensions. The z-axis is perpendicularly aligned to the plane of the display, with the positive z-axis extending towards the viewer. Every sheet of material occupies a single position along the z-axis and has a standard 1dp thickness, equivalent to one pixel of thickness on screens with a pixel density of 160.

Within the material environment, virtual lights illuminate the scene. Key lights create directional shadows, while ambient light creates soft shadows from all angles.



Android gives you everything you need to build best-in-class app experiences. It gives you a single application model that lets you deploy your apps broadly to hundreds of millions of users across a wide range of devices—from phones to tablets and beyond.

Android also gives you tools for creating apps that look great and take advantage of the hardware capabilities available on each device. It automatically adapts your UI to look its best on each device, while giving you as much control as you want over your UI on different device types.

For example, you can create a single app binary that's optimized for both phone and tablet form factors. You declare your UI in lightweight sets of XML resources, one set for parts of the UI that are common to all form factors and other sets for optimizations specific to phones or tablets. At runtime, Android applies the correct resource sets based on its screen size, density, locale, and so on.

As an open marketplace, Google Play puts you in control of how you sell your products. You can publish whenever you want, as often as you want, and to the customers you want. You can distribute broadly to all markets and devices or focus on specific segments, devices, or ranges of hardware capabilities.

You can monetize in the way that works best for your business—priced or free, with in-app products or subscriptions—for highest engagement and revenues. You also have complete control of the pricing for your apps and in-app products and can set or change prices in any supported currency at any time.

Beyond growing your customer base, Google Play helps you build visibility and engagement across your apps and brand. As your apps rise in popularity, Google Play gives them higher

placement in weekly "top" charts and rankings, and for the best apps promotional slots in curated collections.

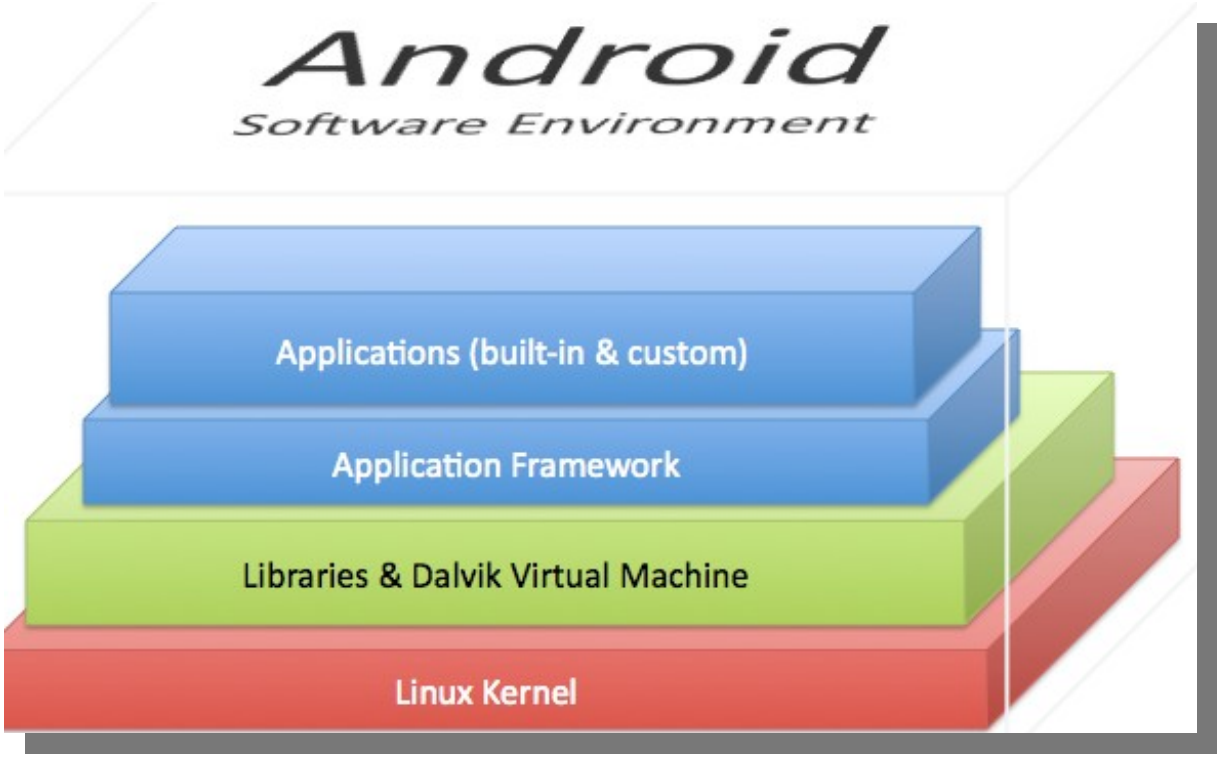
Preinstalled on hundreds of millions of Android devices around the world, Google Play can be a growth engine for your business.

Every day more than 1 million new Android devices are activated worldwide.

Easily optimize a single binary for phones, tablets, and other devices.

1.5 billion downloads a month and growing. Get your apps in front of millions of users at Google's scale.

Android software snvironment sontains a Linux Kernel, Libraries, Dalvik Virtual Machine, Application Framework and built-in and custom applications.



Android is a software environment and not a hardware platform, which includes an OS, built on Linux kernel based OS hosting the Dalvik Virtual machine. The Dalvik Virtual machine runs Android Applications as instances of the virtual machine. Android contains a rich user interface, application framework, Java Class libraries and multimedia support. Android also comes with built in applications containing features such as short message service functionality (messaging), phone capabilities and an address book (contacts).

Dalvik Virtual Machine and Android Applications

Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently.

The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimised for minimal memory footprint.

The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.

The Development Environment

Starting out in Android development can be a daunting task – there's so much information out there, so many tutorials and so many resources it can be hard to navigate.
Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in Java programming language using the Android software development kit (SDK), but other development environments are also available.

Android SDK

A software development kit (SDK or devkit) is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform. To enrich applications with advanced functionalities, advertisements, push notifications and more, most app developers implement specific software development kits. Some SDKs are critical for developing an iOS/Android app. For example, the development of an Android application requires an SDK with Java, for iOS apps an iOS SDK with Swift, and for MS Windows the .NET Framework SDK with .NET. There are also SDKs that are installed in apps to provide analytics and data about activity. Prominent examples include Google, InMobi and Facebook.

Android Studio IDE

Android Studio is the official integrated development environment (IDE) for the Android platform.

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.[6]The first stable build was released in December 2014.

Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development.[8] It is available for download on Windows, macOS and Linux,[9][10] and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

Features

New features are expected to be rolled out with each release of Android Studio. The following features are provided in the current stable version.

- Gradle-based build support
- Android-specific refactoring and quick fixes

- Lint tools to catch performance, usability, version compatibility and other problems
- ProGuard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps

System Requirements

Windows

- Microsoft® Windows® 7/8/10 (32- or 64-bit)
- 6 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- For accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

Mac

- Mac® OS X® 10.10 (Yosemite) or higher, up to 10.12 (macOS Sierra)
- 6 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

Linux

- GNOME or KDE desktop

Tested on Ubuntu® 14.04 LTS, Trusty Tahr (64-bit distribution capable of running 32-bit applications)

- 64-bit distribution capable of running 32-bit applications
- GNU C Library (glibc) 2.19 or later
- 6 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- For accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality, or AMD processor with support for AMD Virtualization™ (AMD-V™).

Configure Your Build

The Android build system compiles app resources and source code, and packages them into APKs that you can test, deploy, sign, and distribute. Android Studio uses Gradle, an advanced build toolkit, to automate and manage the build process, while allowing you to define flexible custom build configurations. Each build configuration can define its own set of code and resources, while reusing the parts common to all versions of your app. The Android plugin for Gradle works with the build toolkit to provide processes and configurable settings that are specific to building and testing Android applications.

Gradle and the Android plugin run independent of Android Studio. This means that you can build your Android apps from within Android Studio, the command line on your machine, or on machines where Android Studio is not installed (such as continuous integration servers). If you are not using Android Studio, you can learn how to build and run your app from the command line. The output of the build is the same whether you are building a project from the command line, on a remote machine, or using Android Studio.

The Build Process

The build process involves many tools and processes that convert your project into an Android Application Package (APK). The build process is very flexible, so it's useful to understand some of what is happening under the hood.
The build process for a typical Android app module, as shown in figure 1, follows these general steps:

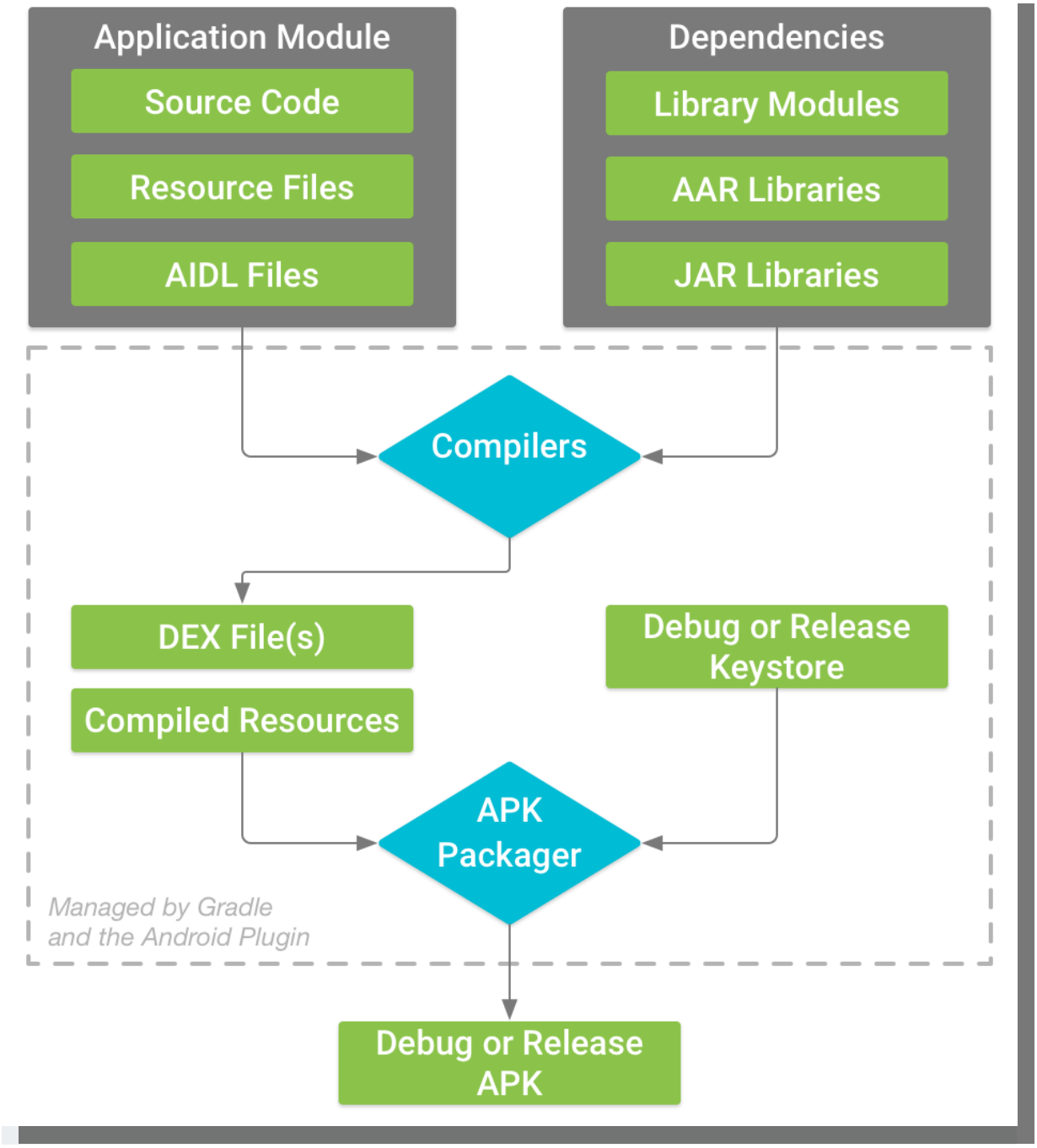
1. The compilers convert your source code into DEX (Dalvik Executable) files, which include the bytecode that runs on Android devices, and everything else into compiled resources.
2. The APK Packager combines the DEX files and compiled resources into a single APK. Before your app can be installed and deployed onto an Android device, however, the APK must be signed.
3. The APK Packager signs your APK using either the debug or release keystore:

a. If you are building a debug version of your app, that is, an app you intend only for testing and profiling, the packager signs your app with the debug keystore. Android Studio automatically configures new projects with a debug keystore.

b. If you are building a release version of your app that you intend to release externally, the packager signs your app with the release keystore. To create a release keystore, read about signing your app in Android Studio.

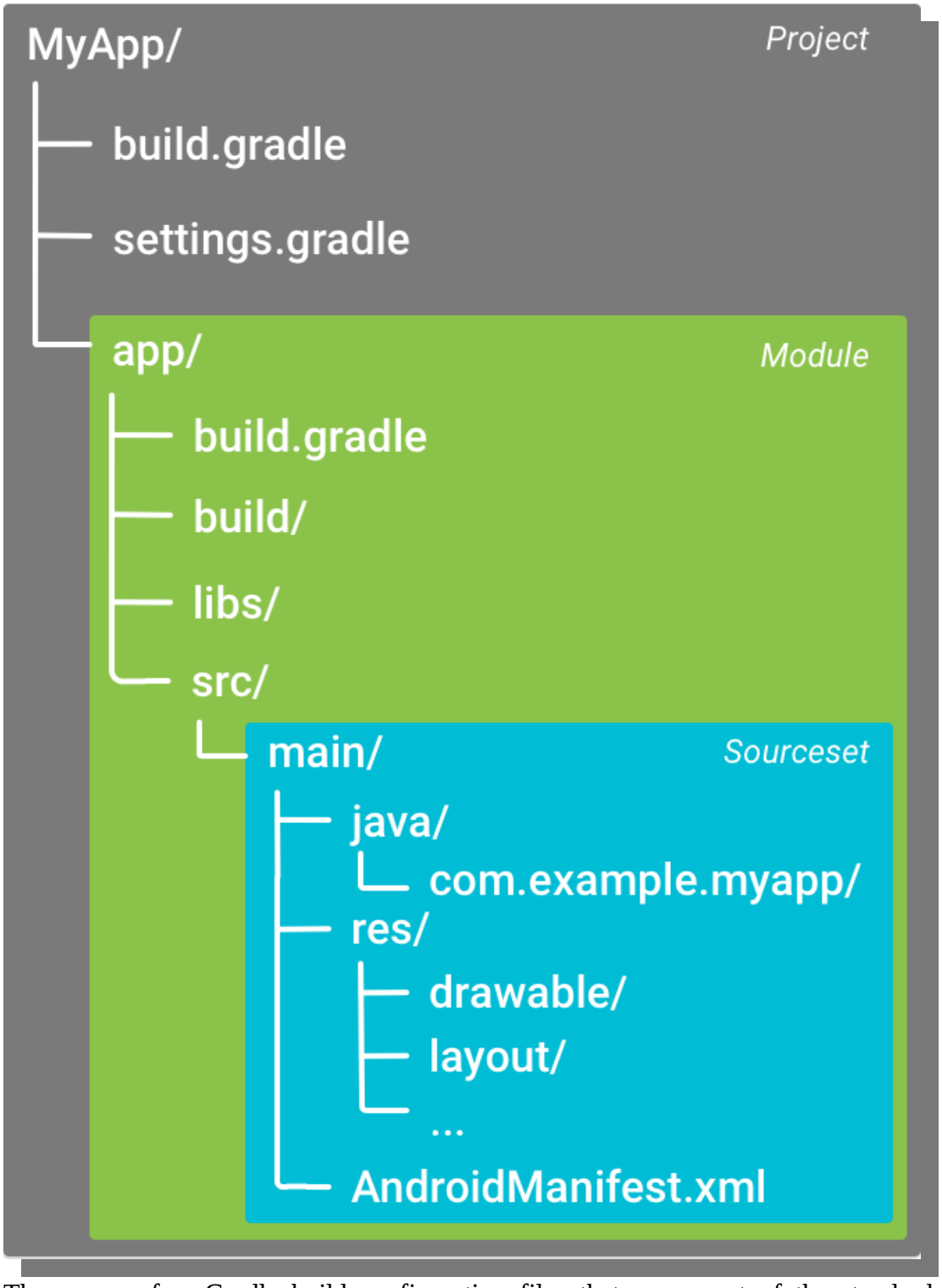
4. Before generating your final APK, the packager uses the zipalign tool to optimize your app to use less memory when running on a device.

At the end of the build process, you have either a debug APK or release APK of your app that you can use to deploy, test, or release to external users.



Build Configuration Files

When starting a new project, Android Studio automatically creates some of these files for you and populates them based on *sensible defaults*.



There are a few Gradle build configuration files that are a part of the standard project structure for an Android app. Before you can start configuring your build, it is important to understand the scope and purpose of each of these files, and the basic DSL elements they should define.

Android Security

Security on watch for your apps

Even after you've installed an app, built-in software regularly scans your device to ensure that the app is behaving. If the app steps out of line, you'll be notified and Google Play can automatically block it in a flash.

Working together to keep you safe

When you shop at a merchant, Android Pay doesn't send your actual credit or debit card number with your payment. Instead we use a virtual account number to represent your account information — so your card details stay safe.

Safe. Secure. Sandboxed.

Just like the walls of a sandbox keep sand from getting out, each Android app is housed in a virtual sandbox that helps keep your personal data safe. This way, apps that you install won't be able to access information like photos or your location unless you give them permission. It's your privacy, so we put you in control.

Protect your phone, even if you lose it

Anybody can lose their phone, but rest assured. With Android Device Manager, you can remotely locate any lost device associated with your Google account, all while keeping your data safe and sound. You can even set a lock screen or erase all data on your device if it's stolen or lost for good. Crisis averted.

Set the perfect password: your fingerprint

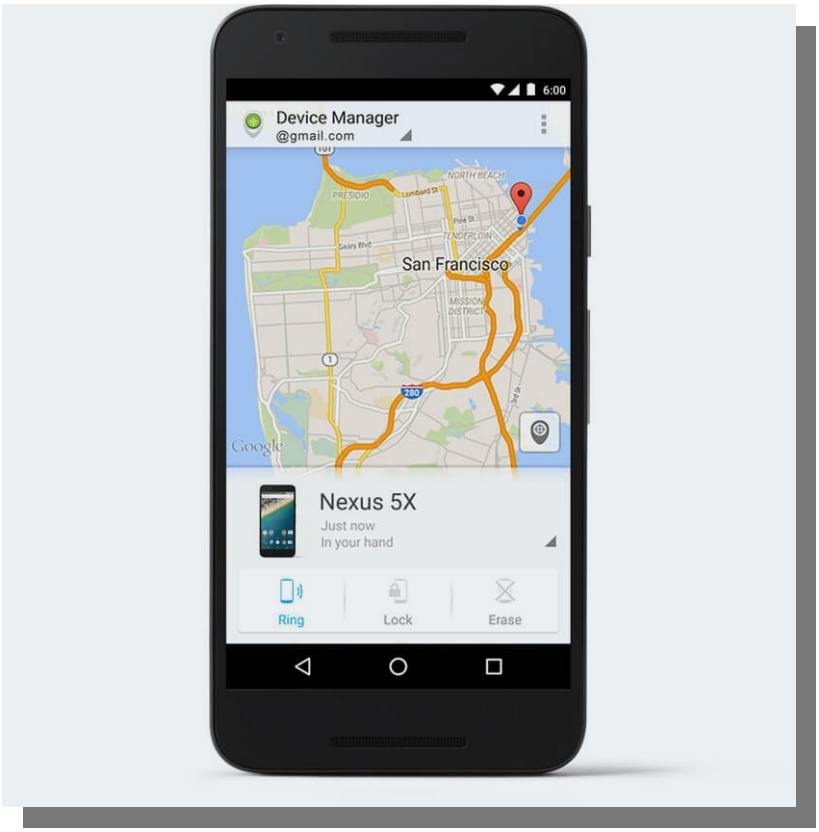
On compatible devices with Android Marshmallow, your fingerprint can unlock your phone, let you securely sign into apps and let you make purchases on Android Pay and the Play Store. Best of all, protecting your device takes just the tap of your finger.

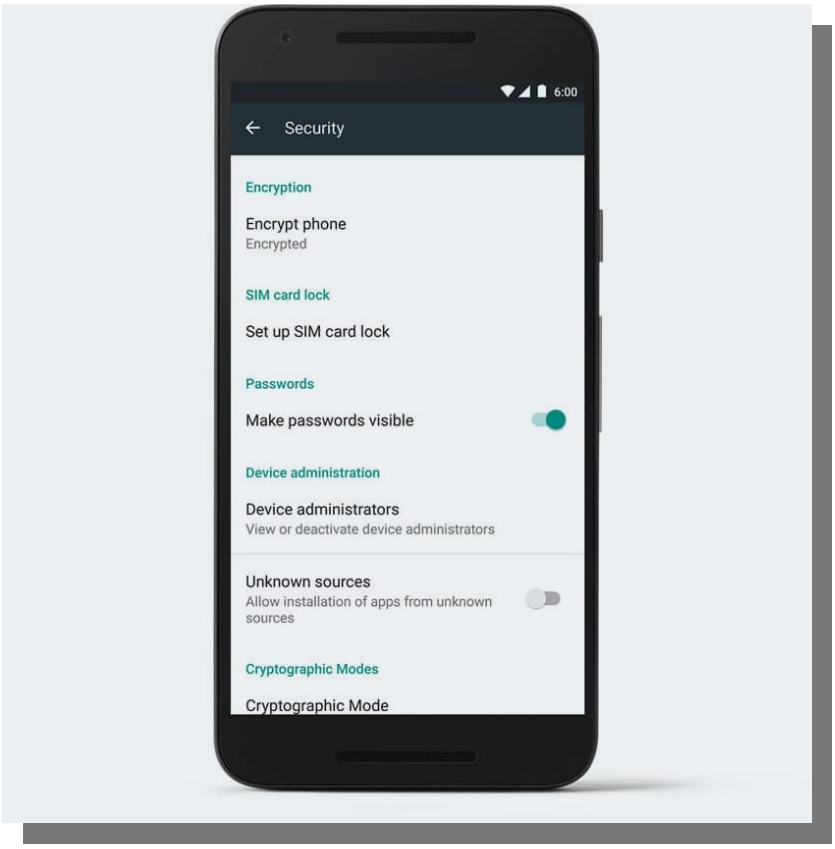
Built-in encryption

With full-disk encryption, all data that exists on your device is fully protected. And any new data, such as emails and text messages, are automatically encrypted when stored locally on your device.

Tested multiple ways before apps see the light of Play

All Android apps undergo rigorous security testing before appearing in the Google Play Store. We vet every app developer in Google Play and suspend those who violate our policies. So even before you install an app, you know that we've checked that it's safe.





Android Easy to Use

For some users, especially older users, smartphones are nothing less than an intimidating device. The same is true for iOS or Android, but for the latter, the customization options can appear to complicate things even more.

Luckily, Android users have the advantage of being able install launchers or customization packs to change how a device looks and works.

Smartphones built specifically for seniors or older.

Android developed OS that is intended to help the elderly with their smartphone needs, by not only providing an interface that is easy to see and navigate, but also offering medical and emergency services that are easily accessible from the device. The company recently released a new smartphone that offers all this and more.

Android Performance

Android Performance Patterns outlining a number of performance issues developers stumble across when creating applications for Android, along with advice on dealing with them which we will present in summary.

Rendering Performance

According to Colt McAnlis, presenter of the video series, improper rendering is the source of most performance issues on Android. If an activity needs more than 16ms to prepare the rendering of the next frame on the screen the system will drop the respective frame displaying instead the previous one. When frames are dropped the user may have an unpleasant experience using the application noticing scrolling that is not smooth or laggy transitions.

Some of the reasons for rendering performance issues are: redrawing a large view hierarchy, drawing many objects on top of each other, or repeated animations. McAnlis recommends a number of tools for investigating with such problems: Hierarchy Viewer, Traceview, Profile GPU Rendering, Debug GPU Overdraw, and GPU View Updates.
It is recommended to minimize the number of invalidations and layouts. The former are visualized with the Developer Option Show GPU View Updates, while the later can be investigated with Hierarchy Viewer.

Overdraw

Overdraw measures the number of times a certain pixel is drawn in a single frame. It is desirable for this counter to be 1, but this is not always possible. The Debug GPU Overdraw tool can show the areas that are mostly redrawn so the developer can know what to optimize.

Profile GPU Rendering

Accessible from the Developer Options setting, this tool generates graphs for all the Android activities active on the screen, usually the notification and navigation bars plus the active application. It is desirable to see frame rates under the 16ms threshold. The performance graph contains timing information for all major activities involved in rendering: updating a display list, executing a display list and processing glSwapBuffers.

60 FPS

The desired rate of frames per second is minimum 60 fps which allows complex animations to be perceived well by the human eye. It is important to do 60 fps and avoid variations.

Custom Views

When the onDraw() method is overridden in order to create a custom view, the rendering process misses some Android overdraw optimizations such not drawing components that are completely covered by others. To minimize the lack of overdraw optimization, it is recommended to use canvas.clipRect() to specify the exact area that is custom drawn. Also, quickReject() should be used to determine if a region is outside the clipped rectangle, avoiding spending time drawing it when this is not necessary.

Memory Churn

Repeatedly allocating memory for lots of small memory objects which later are discarded forces the garbage collector to intervene multiple times, taking time from the 16ms window and possibly leading to frame dropping. Using the Android Studio Memory Monitor one can visualize GC activity and determine if there is too much garbage collection. The Allocation Tracker can then be used to determine where the memory objects are coming from. Fixing the related code may involve avoiding some memory allocations or moving them outside loops. Also, objects should not be allocated inside onDraw() because the method is called many times a second. When the application really needs a larger number of objects that are discarded shortly after, it is recommended to use a pool of objects that are created once and reused, thus avoiding the GC.

Memory Leaks

Memory leaks make GC take longer to complete which may impact the frame rate. To make sure an Android activity does not leak memory after the user leaves it, McAnlis recommends creating a blank activity with no or very little memory consumption, transition to it and force a garbage collection. Investigate memory consumption with the Heap and the Allocation Tracker tools in Android Studio before and after the transition to find out if the activity leaks memory or not.

Battery Consumption

According to a Purdue and Microsoft research paper (PDF), about 70% of the battery consumption of several high profile apps went to third party advertising modules which performed data analytics, location discovery and advertisement downloads. Only 30% of the battery went to the actual activity of the application. Google advises developers to be careful with battery consumption because this is on top of users' concerns list. Top recommendation is avoiding waking a device from sleep unless absolutely necessary. If it has to be awaken, it is recommended using WakeLock with a timeout to make sure the device goes back to sleep in case something went wrong with the app. Another idea is to use AlarmManager.setInexactRepeating() to combine the wake up with another activity to save battery.

Battery consumption can be addressed by deferring some CPU intensive operations for a time when the device is connected to the charger, connected to WiFi or to combine multiple jobs in one device wake up. This can be done with the JobScheduler API which enables deferring some jobs to a later time. It is also recommended to perform network connections with care because after sending a network request and receiving a response, the device is kept awake for at least another 5 seconds just in case another data packet arrives from the server.

McAnlis also suggests using the consumption details offered by Settings|Battery and the Battery Historian tool to monitor how an app consumes battery over time.

Firebase

“The tools and infrastructure you need to build better apps and grow successful businesses”.

Firebase is a mobile and web application platform with tools and infrastructure designed to help developers build high-quality apps. Firebase is made up of complementary features that developers can mix-and-match to fit their needs. The team is based in San Francisco and Mountain View, California. The company was founded in 2011 by Andrew Lee and James Tamplin. Firebase's initial product was a realtime database, which provides an API that allows developers to store and sync data across multiple clients. Over time, it has expanded its product line to become a full suite for app development. The company was acquired by Google in October 2014 and a significant number of new features were featured in May 2016.

Firebase evolved from Envolve, a prior startup founded by Tamplin and Lee in 2011. Envolve provided developers an API that let them integrate online chat into their websites. After releasing the chat service, Tamplin and Lee found that the service was being used to pass application data that wasn't chat messages. Developers were using Envolve to sync application data such as game state in realtime across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it, founding Firebase as a separate company in April 2012. Firebase raised $1.4 million in seed funding in May 2012 from Flybridge Capital Partners, Greylock Partners, NEA and others. The company raised $5.6 million in Series A funding from Union Square Ventures and Flybridge Capital Partners in June 2013. On 21 October 2014, Firebase announced it had been acquired by Google for an undisclosed amount. On October 13, 2015 Google acquired Divshot to merge it with Firebase team, for an undisclosed amount. Before the acquisition, Divshot had raised $1.18 million in two rounds of funding, according to TechCrunch. Since the acquisition Firebase has grown inside Google and expanded their services to become a unified platform for mobile developers. Firebase now integrates with various other Google services to offer broader products and scale for developers. The vision of Firebase stays the same and aim to help developers build better apps and grow successful businesses.

On January 18, 2017, Google announced that it signed an agreement to acquire Fabric and Crashlytics from Twitter, and that those services would join the Firebase team.

Firebase is a unified mobile development platform from Google. With a single suite of tools, app developers & marketers have the resources to develop great apps, grow and re-engage users, and monetize their app. There are 15 features within Firebase, that work across Android, iOS and web. Where applicable, these features work seamlessly together for an intuitive, powerful developer experience.

Features

Firebase helps you to develop high-quality apps, grow your base, and earn more money. Each feature works independently, and they work even better together.

- ### Analytics

  At the heart of Firebase is Firebase Analytics, a free and unlimited analytics solution. See user behaviour and measure attribution from a single dashboard.

- Unlimited reporting of 500 event types, each with up to 25 attributes
- One dashboard to view user behaviour and cross-network campaign performance
- Demographic segmentation, including age, gender, and location, available out-of-the-box
- Export raw data to BigQuery for custom querying
- The Firebase Analytics Partner database includes leading mobile advertising technology platforms, which have been validated to measure and optimize app campaign performance.

- **Cloud Messaging**

  Deliver and receive messages across platforms reliably.
  Formerly known as Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) is a cross-platform — Android, iOS, and Web — solution that lets you reliably deliver and receive messages and notifications at no cost.

  - Send unlimited upstream/downstream messages
  - Send messages to individual devices or a user segment
  - Handle all aspects of queueing and delivery
  - Optimize for battery efficiency

- **Authentication**

  Implement a complete authentication system that supports email & password, Facebook, Twitter, GitHub and Google Sign-In. Already have an account system? Easily integrate it with our custom auth service to give your users secure access to many of Firebase's other features.

  - Flexible SDKs on top of reliable Google infrastructure
  - Optional, out-of-the-box authentication UI optimized to give your users the best experience
  - Advanced functionality like email verification, anonymous accounts, and account linking

- **Realtime database**

  A cloud-hosted NoSQL database. Data is stored as JSON, synced across connected devices in milliseconds, and available when your app goes offline.

  - Intuitive and easy-to-use API
  - Remains responsive regardless of network latency or Internet connectivity so your Firebase app works offline. Data synchronizes when connectivity returns
  - Handles the complexity of realtime synchronization and provides flexible conflict resolution
  - Accessible directly from client SDKs, or from your server with the REST API

- **Test lab**

  Test in the lab, not on your users

  - Generate detailed reports and screenshots to help identify bugs
  - Run custom test scripts on hundreds of device configurations
  - Supplement your existing workflow through integration with Android Studio, command-line tools, and Web-based consoles

- **Crash Reporting**

  Receive actionable information on stability issues after you publish your app.

  - Prioritize crashes by frequency and impact
  - Comprehensive data surrounding each crash, including device characteristics, device circumstances, a stack trace, and more
  - Reliably collect crashes that occur while the device is online or offline

- Ability to measure the impact of a crash on user behaviour via Firebase Analytics

- **Cloud Storage**

  Store and serve content with ease.

  Store and serve user-generated content like images, audio, and video directly in your mobile app, using the Firebase SDKs. Power your app with the same technology that powers Snapchat and Spotify.

  - Resumable uploads and downloads, resilient to changes in network connectivity
  - Secure client side authorization, integrated with Firebase Authentication
  - Scales to exabytes of storage for your app
  - All data is accessible from the Cloud Storage server SDK, as well as the JSON and S3-compatible XML APIs.

- **Cloud Code**

  Run your mobile backend code without managing servers.

  Cloud Functions for Firebase allows you to extend and connect Firebase and cloud services with code. You can run your mobile backend code that automatically responds to events triggered by Firebase features and HTTPS requests without the need to manage and scale your own servers.

  - Cloud Functions respond to events generated by other Firebase project features, such as Authentication, Realtime Database, Cloud Storage and more
  - Cloud Functions are fully insulated from the client so you can be sure they are private and secure
  - Deploying your code to our servers requires just one command.

- **Hosting**

  Deliver web content faster

  Production-grade hosting for developers. Deploy Web and mobile Web apps to a global content-delivery network (CDN) with a single command.

  - Automatically provisioned SSL certificate
  - Blazing-fast content worldwide
  - Support for client-side routing
  - Atomic deploys and one-click rollbacks

- **Cloud Function**

  Run your mobile backend code without managing servers.

Cloud Functions for Firebase allows you to extend and connect Firebase and cloud services with code. You can run your mobile backend code that automatically responds to events triggered by Firebase features and HTTPS requests without the need to manage and scale your own servers.

- Cloud Functions respond to events generated by other Firebase project features, such as Authentication, Realtime Database, Cloud Storage and more
- Cloud Functions are fully insulated from the client so you can be sure they are private and secure
- Deploying your code to our servers requires just one command

- **Crash Reporting**

  Keep your app stable.
  Receive actionable information on stability issues after you publish your app.

  - Prioritize crashes by frequency and impact
  - Comprehensive data surrounding each crash, including device characteristics, device circumstances, a stack trace, and more
  - Reliably collect crashes that occur while the device is online or offline
  - Ability to measure the impact of a crash on user behavior via Firebase Analytics

Grow

Acquire and engage the right users at the right time. Take the guesswork out of growth.

- **Notifications**

  Engage with users at the right moment
  Easily manage notification campaigns. Schedule and send messages to engage the right users at the most relevant time.

  - Send free and unlimited notifications across Android, and iOS.
  - Send messages and analyze effectiveness in one dashboard without writing any code
  - Integrate with Firebase Analytics to deliver messages to a user segment

- **Remote Config**

  Customize your app on the fly.
  Update your app without deploying a new version. Quickly deliver the right experience to the right users.

  - Modify your app without a new production deployment
  - Customize content for different Firebase Analytics audiences and measure results

- Roll out features gradually and monitor the impact

- **App Indexing**

  Customize your app on the fly

  Update your app without deploying a new version. Quickly deliver the right experience to the right users.

  - Modify your app without a new production deployment
  - Customize content for different Firebase Analytics audiences and measure results
  - Roll out features gradually and monitor the impact

- **Dynamic Links**

  Send users to the right place inside your app.

  Dynamic Links are smart URLs that dynamically change behavior to provide the best experience across different platforms. Dynamic Links can survive the app install process and take users to relevant content whether they're a brand-new user or a longtime customer.

  - Improve acquisition and engagement by bringing users directly to content that they were originally searching for, whether they have your app installed or not
  - Delight new users with personalized promotions and messages after install

- **Invites**

  Empower your users to share your app.

  Out-of-the-box solution for app referrals and sharing. Let your existing users easily share your app, or their favorite in-app content, via email or SMS. Use in conjunction with promotions to increase acquisition and retention.

  - Invite the most relevant contacts with smart suggestions
  - Free email and SMS delivery
  - Powered by Firebase Dynamic Links

- **AdWords**

  Acquire users with the reach of Google.

  Automatically link AdWords to a user segment you define in Firebase Analytics. Improve ad targeting and optimize your campaign performance.

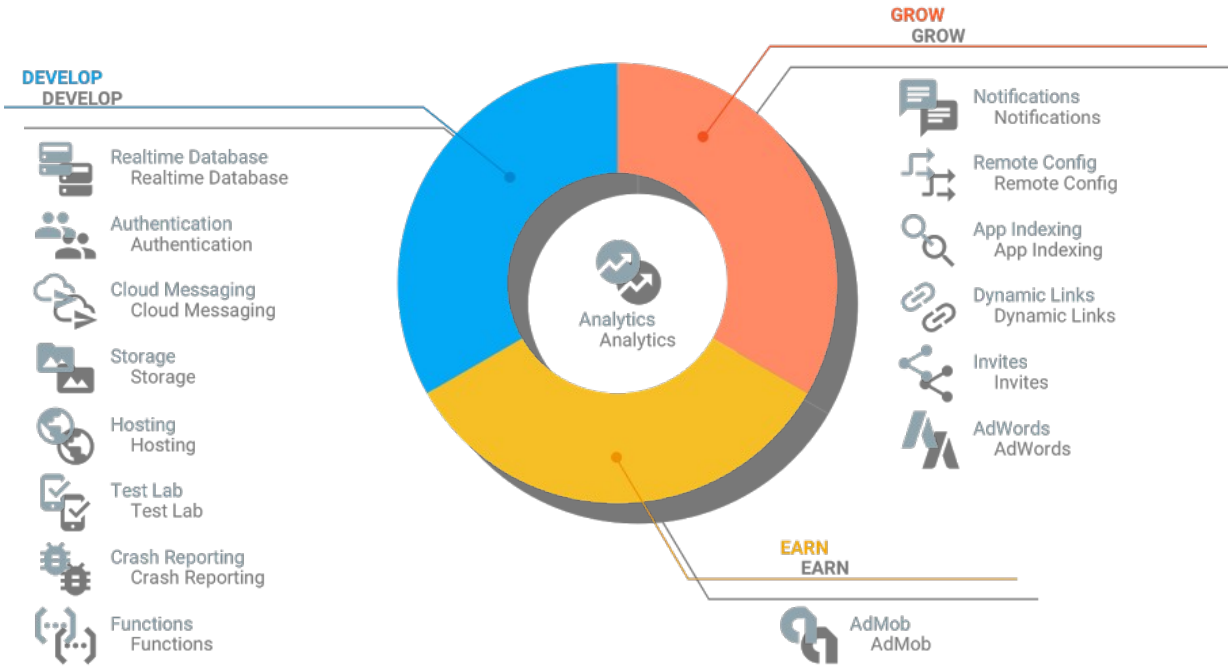
  - Conversion tracking for first opens and in-app events without implementing any additional SDKs
  - Cross-network attribution measurement and LTV in one dashboard
  - Show ads to users based on user segments from Firebase Analytics

Earn

Earn money by displaying engaging ads to a global audience.

- **AdMob**
  Monetize through engaging ads.
  Monetize your app while giving your users a great experience.

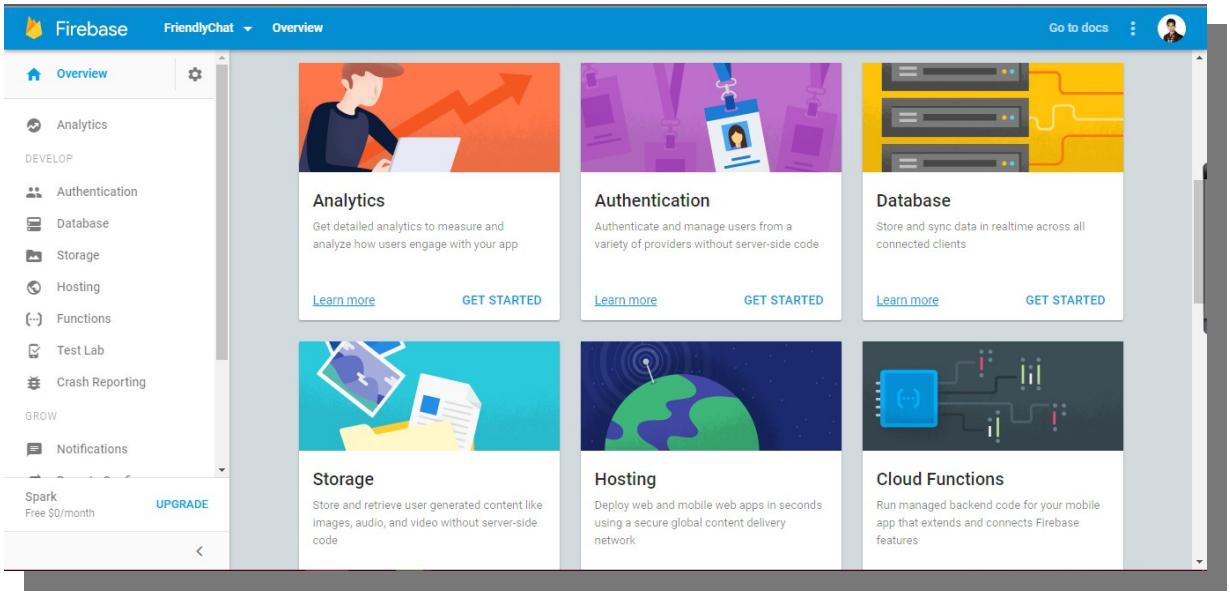
    - Show ads from millions of Google advertisers competing in real time
    - Choose a format to suit your app, including banner, video and native ads
    - Work with more than 40 top ad networks using AdMob Mediation
    - Cross-promote between your apps for free with AdMob house ads



Disadvantages Of Firebase

As everything has advantages and as well as has disadvantages so firebase has disadvantages which are given below

- Have to build indexes manually
- Might need to build “event log” physically as well (in different sub-tree?
- Implementation of Leftovers API could be difficult on inserted platforms
- Data validation guidelines do not support intricate objects straight (you’d need to validate specific child nodes independently).


Firebase
FriendlyChat
Overview
Go to docs
Overview
Analytics
DEVELOP
Authentication
Database
Storage
Hosting
Functions
Test Lab
Crash Reporting
GROW
Notifications
Spark
Free $0/month
UPGRADE
Analytics
Get detailed analytics to measure and analyze how users engage with your app
Learn more
GET STARTED
Authentication
Authenticate and manage users from a variety of providers without server-side code
Learn more
GET STARTED
Database
Store and sync data in realtime across all connected clients
Learn more
GET STARTED
Storage
Store and retrieve user generated content like images, audio, and video without server-side code
Hosting
Deploy web and mobile web apps in seconds using a secure global content delivery network
Cloud Functions
Run managed backend code for your mobile app that extends and connects Firebase features