**Third Year B. Tech., Sem VI 2021-22**

**Software Engineering Tools Lab**

**Assignment No. 2**

**(Module 2- Software Development Frameworks)**

**PRN:2019BTECS00047**

**PRN:2019BTECS00049**

Que No.1 For **Android SDK** provide the answers for below questions.

1. **Original author :**

Android was created by the [Open Handset Alliance](https://en.wikipedia.org/wiki/Open_Handset_Alliance) which is led by Google.

1. **Developers :**

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

**3. Initial release :**

On October 27,2009 the Android SDK was released , based on linux kernel 2.6.29 and codenamed Éclair.

**4. Stable release :**

A preview release of the Android SDK was released on September 2017.

**5. Preview release :**

A preview release of the Android SDK was released on November 12, 2007.

**6. Repository (with cloud support ) :**

https://android.googlesource.com/

In Android Studio, you can add new projects to [Cloud Source Repositories](https://cloud.google.com/tools/cloud-repositories/docs), and check out existing projects. Currently only **Git** is supported.

To upload your source to Cloud Source Repositories from Android Studio:

1. In Android Studio, open the project you want to add to Cloud Source Repositories.
2. Choose VCS > Import into Version Control > Upload Project to Google Cloud.
3. In the displayed dialog, select the Google Cloud Console project ID associated with the project.
4. Click Upload.

**7. Written in (Languages) :**

### 1. [Java](https://www.geeksforgeeks.org/java/)

Firstly Java was the official language for Android App Development (but now it was replaced by Kotlin) and consequently, it is the most used language as well. Many of the apps in the Play Store are built with Java, and it is also the most supported language by Google. In addition to all this, Java has a great online community for support in case of any problems (And trust me, there will be problems!).

However, Java is a complicated language for a beginner to use as it contains complex topics like constructors, null pointer exceptions, concurrency, checked exceptions, etc. Also, The Android Software Development Kit (SDK) increases the complexity to a new level!

All in all, Java is a great language to experience the full joys of Android App Development. However, it may be a little complex with beginners who would prefer to start with something easier and then return to it.

### 2. [Kotlin](https://www.geeksforgeeks.org/kotlin-language-android-now-official-google/)

Now Kotlin is the official language for Android App Development declared by Google since 2019. Kotlin is a cross-platform programming language that may be used as an alternative to Java for Android App Development. It has also been introduced as a secondary “official” Java language in 2017. Kotlin can interoperate with Java and it runs on the Java Virtual Machine.

The only sizable difference is that Kotlin removes the superfluous features of Java such as null pointer exceptions. It also removes the necessity of ending every line with a semicolon. In short, Kotlin is much simpler for beginners to try as compared to Java and it can also be used as an “entry point” for Android App Development.

### 3. [C++](https://www.geeksforgeeks.org/c-plus-plus/)

C++ can be used for Android App Development using the Android Native Development Kit(NDK). However, an app cannot be created totally using C++ and the NDK is used to implement parts of the app in C++ native code. This helps in using C++ code libraries for the app as required.

While C++ is useful for Android App Development in some cases, it is much more difficult to set up and is much less flexible. It may also lead to more bugs because of the increased complexity. So, it is better to use Java as compared to C++ as it does not provide enough gain to offset the efforts required.

### 4. [C#](https://www.geeksforgeeks.org/csharp-programming-language/)

C# is quite similar to Java and so it is ideal for Android App Development. Like Java, C# also implements garbage collection so there are fewer chances of memory leaks. And C# also has a cleaner and simpler syntax than Java which makes coding with it comparatively easier.

Earlier, the biggest drawback of C# was that it could run only on Windows systems as it used the .NET Framework. However, this problem was handled by Xamarin. Android (formerly Mono for Android) is a cross-platform implementation of the Common Language Infrastructure. Now, Xamarin. Android tools can be used to write native Android apps and share the code across multiple platforms.

### 5. [Python](https://www.geeksforgeeks.org/python-programming-language/)

Python can be used for Android App Development even though Android doesn’t support native Python development. This can be done using various tools that convert the Python apps into Android Packages that can run on Android devices.

An example of this is Kivy that is an open-source Python library used for developing mobile apps. It supports Android and also encourages rapid app development (which is a win-win situation according to me!). However, a downside to this is that there won’t be native benefits for Kivy as it isn’t natively supported.

### 6. [HTML](https://www.geeksforgeeks.org/html-tutorials/), [CSS](https://www.geeksforgeeks.org/css-tutorials/), [JavaScript](https://www.geeksforgeeks.org/javascript-tutorial/)

Android apps can be created using HTML, CSS, and JavaScript using the Adobe PhoneGap framework that is powered by Apache Cordova. The PhoneGap framework basically allows the usage of web development skills to create hybrid apps that are shown through “WebView” but are packaged like an app.

While the Adobe PhoneGap framework is enough for basic tasks in the realm of Android App Development, it hardly requires much programming except for JavaScript. And since it needs a lot of work to even create a decent app, it is better to use the other languages in this list if you want to be called a true Android developer(Yes…That’s a thing!)

### 7. [Dart](https://www.geeksforgeeks.org/dart-tutorial/)

Ignoring Dart as a programming language in today’s context would be like ignoring the gorilla in the room (because the elephant is java). Dart is an open-source programming language that powers the Flutter framework, which is getting a lot of traction these days because of its ability to deliver beautiful and performance app for the web, desktop, and mobile in lesser time. The key selling point of dart is that it is designed by Google as a client-optimized language for fast apps on any platform. Dart mainly focuses on making UI development easier for developers with features such as hot-reload, which lets developers see changes instantly while working on the app. Dart is also known for its fast performance, it compiles to ARM and x64 machine code for mobile, desktop, and backend. And to JavaScript for the web apps.

**8. Operating System support :**

**Android** is a [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) based on a modified version of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) and other [open source](https://en.wikipedia.org/wiki/Open-source_software) software, designed primarily for [touchscreen](https://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](https://en.wikipedia.org/wiki/Smartphone) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer). Android is developed by a consortium of developers known as the [Open Handset Alliance](https://en.wikipedia.org/wiki/Open_Handset_Alliance) and commercially sponsored by [Google](https://en.wikipedia.org/wiki/Google).

**9. Platform ,portability :**

Android SDK Platform-Tools is a component for the Android SDK. It includes tools that interface with the Android platform, such as [adb](https://developer.android.com/studio/command-line/adb), [fastboot](https://android.googlesource.com/platform/system/core/+/master/fastboot/" \l "fastboot), and [systrace](https://developer.android.com/topic/performance/tracing/command-line). These tools are required for Android app development. They're also needed if you want to unlock your device bootloader and flash it with a new system image.

Although some new features in these tools are available only for recent versions of Android, the tools are backward compatible, so you need only one version of the SDK Platform-Tools.

AS (Android Studio) itself is portable, however the JDK (Java Development Kit) which is required to run AS, is not. So you still need to install JDK on each computer you use.

To set up gradle for offline, go to: File > Settings > Build, Execution, Deployment > Gradle then check Offline Work .

**10. Available in (Total languages) :**

Available in 100+ Languages

**11. List of languages supported :**

* Kotlin
* C++
* C#
* Python
* HTML , CSS , Javascript
* Dart

**12. Type (Programming tool, integrated development environment etc.) :**

The Android software development kit (SDK) includes different components, including SDK Tools, Build Tools, and Platform Tools. The SDK Tools primarily includes the stock Android emulator, [hierarchy viewer](https://developer.android.com/studio/profile/hierarchy-viewer.html), [SDK manager](https://developer.android.com/studio/intro/update.html), and [ProGuard](https://guides.codepath.com/android/Configuring-ProGuard). The Build Tools primarily include aapt (Android packaging tool to create .APK), dx (Android tool that converts .java files to .dex files). Platform Tools include the [Android debug shell](https://developer.android.com/studio/command-line/adb.html), [sqlite3](https://developer.android.com/studio/command-line/sqlite3.html) and [Systrace](https://developer.android.com/studio/profile/systrace-commandline.html).

**13. Website :**

Official Website : https://www.android.com/

**14. Features :**

### Interface

Android's default user interface is mainly based on [direct manipulation](https://en.wikipedia.org/wiki/Direct_manipulation_interface), using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a [virtual keyboard](https://en.wikipedia.org/wiki/Virtual_keyboard). [Game controllers](https://en.wikipedia.org/wiki/Game_controller) and full-size physical [keyboards](https://en.wikipedia.org/wiki/Computer_keyboard) are supported via [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth) or [USB](https://en.wikipedia.org/wiki/USB). The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide [haptic feedback](https://en.wikipedia.org/wiki/Haptic_technology) to the user. Internal hardware, such as [accelerometers](https://en.wikipedia.org/wiki/Accelerometer), [gyroscopes](https://en.wikipedia.org/wiki/Gyroscope) and [proximity sensors](https://en.wikipedia.org/wiki/Proximity_sensor) are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a [racing game](https://en.wikipedia.org/wiki/Racing_game) by rotating the device, simulating control of a [steering wheel](https://en.wikipedia.org/wiki/Steering_wheel).

#### **Home screen**

Android devices boot to the [home screen](https://en.wikipedia.org/wiki/Home_screen), the primary navigation and information "hub" on Android devices, analogous to the [desktop](https://en.wikipedia.org/wiki/Desktop_metaphor) found on personal computers. Android home screens are typically made up of app icons and [widgets](https://en.wikipedia.org/wiki/Software_widget); app icons launch the associated app, whereas widgets display live, auto-updating content, such as a [weather forecast](https://en.wikipedia.org/wiki/Weather_forecast), the user's email inbox, or a [news ticker](https://en.wikipedia.org/wiki/News_ticker) directly on the home screen. A home screen may be made up of several pages, between which the user can swipe back and forth. Third-party apps available on [Google Play](https://en.wikipedia.org/wiki/Google_Play) and other app stores can extensively re-[theme](https://en.wikipedia.org/wiki/Theme_(computing)) the home screen, and even mimic the look of other operating systems, such as [Windows Phone](https://en.wikipedia.org/wiki/Windows_Phone).[[94]](https://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-94) Most manufacturers customize the look and features of their Android devices to differentiate themselves from their competitors.

#### **Status bar**

Along the top of the screen is a status bar, showing information about the device and its connectivity. This status bar can be pulled (swiped) down from to reveal a notification screen where apps display important information or updates, as well as quick access to system controls and toggles such as display brightness, connectivity settings ([WiFi](https://en.wikipedia.org/wiki/WiFi" \o "WiFi), [Bluetooth](https://en.wikipedia.org/wiki/Bluetooth), cellular data), audio mode, and [flashlight](https://en.wikipedia.org/wiki/Flashlight). Vendors may implement extended settings such as the ability to adjust the flashlight brightness.

#### **Notifications**

Notifications are "short, timely, and relevant information about your app when it's not in use", and when tapped, users are directed to a screen inside the app relating to the notification. Beginning with [Android 4.1 "Jelly Bean"](https://en.wikipedia.org/wiki/Android_Jelly_Bean), "expandable notifications" allow the user to tap an icon on the notification in order for it to expand and display more information and possible app actions right from the notification

#### **App lists**

An "All Apps" screen lists all installed applications, with the ability for users to drag an app from the list onto the home screen. The app list may be accessed using a gesture or a button, depending on the Android version. A "Recents" screen, also known as "Overview", lets users switch between recently used apps.

The recent list may appear side-by-side or overlapping, depending on the Android version and manufacturer.

#### **Split-screen view**

Native support for split screen view has been added in stock Android version 7.0 *Nougat*.

The earliest vendor-customized Android-based smartphones known to have featured a split-screen view mode are the 2012 [Samsung Galaxy S3](https://en.wikipedia.org/wiki/Samsung_Galaxy_S3) and [Note 2](https://en.wikipedia.org/wiki/Samsung_Galaxy_Note_II), the former of which received this feature with the *premium suite* upgrade delivered in [TouchWiz](https://en.wikipedia.org/wiki/TouchWiz) with Android 4.1 Jelly Bean.

#### **Charging while powered off**

When connecting or disconnecting charging power and when shortly actuating the power button or home button, all while the device is powered off, a visual battery meter whose appearance varies among vendors appears on the screen, allowing the user to quickly assess the charge status of a powered-off without having to boot it up first. Some display the battery percentage.

#### **Audio-coupled haptic effect**

Since stock Android version 12, released early 2021, synchronous vibration can be set to complement audio. Such feature initially existed under the name "Auto Haptic" on the Android-based 2012 [Samsung Galaxy S III](https://en.wikipedia.org/wiki/Samsung_Galaxy_S_III), released with a vendor-modified ([TouchWiz](https://en.wikipedia.org/wiki/TouchWiz)) installation of Android 4.1 Jelly Bean.

**15. Size (in MB, GB etc.)**

Android SDK (Software Development Kit) (**about 5 GB**) for developing and testing Android apps.

**16. Privacy and Security :**

### Common security threats

Research from security company [Trend Micro](https://en.wikipedia.org/wiki/Trend_Micro) lists premium service abuse as the most common type of Android malware, where text messages are sent from infected phones to [premium-rate telephone numbers](https://en.wikipedia.org/wiki/Premium-rate_telephone_number) without the consent or even knowledge of the user. Other malware displays unwanted and intrusive advertisements on the device, or sends personal information to unauthorised third parties. Security threats on Android are reportedly growing exponentially; however, Google engineers have argued that the malware and virus threat on Android is being [exaggerated](https://en.wikipedia.org/wiki/Fear,_uncertainty_and_doubt) by security companies for commercial reasons, and have accused the security industry of playing on fears to sell virus protection software to users. Google maintains that dangerous malware is actually extremely rare, and a survey conducted by [F-Secure](https://en.wikipedia.org/wiki/F-Secure) showed that only 0.5% of Android malware reported had come from the Google Play store.

#### **Scope of surveillance by public institutions**

As part of the broader [2013 mass surveillance disclosures](https://en.wikipedia.org/wiki/2013_mass_surveillance_disclosures) it was revealed in September 2013 that the American and British intelligence agencies, the [National Security Agency](https://en.wikipedia.org/wiki/National_Security_Agency) (NSA) and [Government Communications Headquarters](https://en.wikipedia.org/wiki/Government_Communications_Headquarters) (GCHQ), respectively, have access to the user data on iPhone, BlackBerry, and Android devices. They are reportedly able to read almost all smartphone information, including SMS, location, emails, and notes. In January 2014, further reports revealed the intelligence agencies' capabilities to intercept the personal information transmitted across the Internet by social networks and other popular applications such as [*Angry Birds*](https://en.wikipedia.org/wiki/Angry_Birds), which collect personal information of their users for advertising and other commercial reasons. GCHQ has, according to [*The Guardian*](https://en.wikipedia.org/wiki/The_Guardian), a [wiki](https://en.wikipedia.org/wiki/Wiki)-style guide of different apps and advertising networks, and the different data that can be siphoned from each. Later that week, the Finnish Angry Birds developer [Rovio](https://en.wikipedia.org/wiki/Rovio_Entertainment) announced that it was reconsidering its relationships with its advertising platforms in the light of these revelations, and called upon the wider industry to do the same.

#### **Security patches**

In August 2015, Google announced that devices in the [Google Nexus](https://en.wikipedia.org/wiki/Google_Nexus) series would begin to receive monthly security [patches](https://en.wikipedia.org/wiki/Patch_(computing)). Google also wrote that "Nexus devices will continue to receive major updates for at least two years and security patches for the longer of three years from initial availability or 18 months from last sale of the device via the [Google Store](https://en.wikipedia.org/wiki/Google_Store).The following October, researchers at the [University of Cambridge](https://en.wikipedia.org/wiki/University_of_Cambridge) concluded that 87.7% of Android phones in use had known but unpatched [security vulnerabilities](https://en.wikipedia.org/wiki/Security_vulnerabilities) due to lack of updates and support. Ron Amadeo of [*Ars Technica*](https://en.wikipedia.org/wiki/Ars_Technica) wrote also in August 2015 that "Android was originally designed, above all else, to be widely adopted. Google was starting from scratch with zero percent market share, so it was happy to give up control and give everyone a seat at the table in exchange for adoption. Now, though, Android has around 75–80 percent of the worldwide smartphone market—making it not just the world's most popular mobile operating system but arguably the most popular operating system, period. As such, security has become a big issue. Android still uses a software update chain-of-command designed back when the Android ecosystem had zero devices to update, and it just doesn't work".

### Location-tracking

Android smartphones have the ability to report the location of [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) access points, encountered as phone users move around, to build databases containing the physical locations of hundreds of millions of such access points. These databases form electronic maps to locate smartphones, allowing them to run apps like [Foursquare](https://en.wikipedia.org/wiki/Foursquare_City_Guide), [Google Latitude](https://en.wikipedia.org/wiki/Google_Latitude), [Facebook Places](https://en.wikipedia.org/wiki/Facebook_Places), and to deliver location-based ads.[]](https://en.wikipedia.org/wiki/Android_(operating_system)#cite_note-276) Third party monitoring software such as TaintDroid, an academic research-funded project, can, in some cases, detect when personal information is being sent from applications to remote servers.

### Further notable exploits

In 2018, Norwegian security firm Promon has unearthed a serious Android security hole which can be exploited to steal login credentials, access messages, and track location, which could be found in all versions of Android, including [Android 10](https://en.wikipedia.org/wiki/Android_10). The vulnerability came by exploiting a bug in the multitasking system enabling a malicious app to overlay legitimate apps with fake login screens that users are not aware of when handing in security credentials. Users can also be tricked into granting additional permissions to the malicious apps, which later enable them to perform various nefarious activities, including intercepting texts or calls and stealing banking credentials.[*Avast*](https://en.wikipedia.org/wiki/Avast)*Threat Labs* also discovered that many pre-installed apps on several hundred new Android devices contain dangerous malware and [adware](https://en.wikipedia.org/wiki/Adware). Some of the preinstalled malware can commit ad fraud or even take over its host device.

### Technical security features

Android applications run in a [sandbox](https://en.wikipedia.org/wiki/Sandbox_(computer_security)), an isolated area of the system that does not have access to the rest of the system's resources, unless access permissions are explicitly granted by the user when the application is installed, however this may not be possible for pre-installed apps. It is not possible, for example, to turn off the microphone access of the pre-installed camera app without disabling the camera completely. This is valid also in Android versions 7 and 8.

Since February 2012, Google has used its [Google Bouncer](https://en.wikipedia.org/wiki/Google_Bouncer) malware scanner to watch over and scan apps available in the Google Play store. A "Verify Apps" feature was introduced in November 2012, as part of the [Android 4.2 "Jelly Bean"](https://en.wikipedia.org/wiki/Android_Jelly_Bean) operating system version, to scan all apps, both from Google Play and from third-party sources, for malicious behaviour. Originally only doing so during installation, Verify Apps received an update in 2014 to "constantly" scan apps, and in 2017 the feature was made visible to users through a menu in Settings.

**17. Type of software (Open source/License) :**

Android is **as much open-source software** as it ever was.

This clause applies to the SDK binary, not the SDK source code files, and it has been around for years.

The SDK source code, like almost all of Android, is covered by the Apache Software License 2 (ASLv2).

**18. If License- Provide details :**

### 1. Introduction

1.1 The Android Software Development Kit (referred to in the License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of the License Agreement. The License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: https://source.android.com/, as updated from time to time.

1.3 A "compatible implementation" means any Android device that (i) complies with the Android Compatibility Definition document, which can be found at the Android compatibility website (https://source.android.com/compatibility) and which may be updated from time to time; and (ii) successfully passes the Android Compatibility Test Suite (CTS).

1.4 "Google" means Google LLC, organized under the laws of the State of Delaware, USA, and operating under the laws of the USA with principal place of business at 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA.

### 2. Accepting this License Agreement

2.1 In order to use the SDK, you must first agree to the License Agreement. You may not use the SDK if you do not accept the License Agreement.

2.2 By clicking to accept and/or using this SDK, you hereby agree to the terms of the License Agreement.

2.3 You may not use the SDK and may not accept the License Agreement if you are a person barred from receiving the SDK under the laws of the United States or other countries, including the country in which you are resident or from which you use the SDK.

2.4 If you are agreeing to be bound by the License Agreement on behalf of your employer or other entity, you represent and warrant that you have full legal authority to bind your employer or such entity to the License Agreement. If you do not have the requisite authority, you may not accept the License Agreement or use the SDK on behalf of your employer or other entity.

### 3. SDK License from Google

3.1 Subject to the terms of the License Agreement, Google grants you a limited, worldwide, royalty-free, non-assignable, non-exclusive, and non-sublicensable license to use the SDK solely to develop applications for compatible implementations of Android.

3.2 You may not use this SDK to develop applications for other platforms (including non-compatible implementations of Android) or to develop another SDK. You are of course free to develop applications for other platforms, including non-compatible implementations of Android, provided that this SDK is not used for that purpose.

3.3 You agree that Google or third parties own all legal right, title and interest in and to the SDK, including any Intellectual Property Rights that subsist in the SDK. "Intellectual Property Rights" means any and all rights under patent law, copyright law, trade secret law, trademark law, and any and all other proprietary rights. Google reserves all rights not expressly granted to you.

3.4 You may not use the SDK for any purpose not expressly permitted by the License Agreement. Except to the extent required by applicable third party licenses, you may not copy (except for backup purposes), modify, adapt, redistribute, decompile, reverse engineer, disassemble, or create derivative works of the SDK or any part of the SDK.

3.5 Use, reproduction and distribution of components of the SDK licensed under an open source software license are governed solely by the terms of that open source software license and not the License Agreement.

3.6 You agree that the form and nature of the SDK that Google provides may change without prior notice to you and that future versions of the SDK may be incompatible with applications developed on previous versions of the SDK. You agree that Google may stop (permanently or temporarily) providing the SDK (or any features within the SDK) to you or to users generally at Google's sole discretion, without prior notice to you.

3.7 Nothing in the License Agreement gives you a right to use any of Google's trade names, trademarks, service marks, logos, domain names, or other distinctive brand features.

3.8 You agree that you will not remove, obscure, or alter any proprietary rights notices (including copyright and trademark notices) that may be affixed to or contained within the SDK.

### 4. Use of the SDK by You

4.1 Google agrees that it obtains no right, title or interest from you (or your licensors) under the License Agreement in or to any software applications that you develop using the SDK, including any intellectual property rights that subsist in those applications.

4.2 You agree to use the SDK and write applications only for purposes that are permitted by

(a) the License Agreement and

(b) any applicable law, regulation or generally accepted practices or guidelines in the relevant jurisdictions (including any laws regarding the export of data or software to and from the United States or other relevant countries).

4.3 You agree that if you use the SDK to develop applications for general public users, you will protect the privacy and legal rights of those users. If the users provide you with user names, passwords, or other login information or personal information, you must make the users aware that the information will be available to your application, and you must provide legally adequate privacy notice and protection for those users. If your application stores personal or sensitive information provided by users, it must do so securely. If the user provides your application with Google Account information, your application may only use that information to access the user's Google Account when, and for the limited purposes for which, the user has given you permission to do so.

4.4 You agree that you will not engage in any activity with the SDK, including the development or distribution of an application, that interferes with, disrupts, damages, or accesses in an unauthorized manner the servers, networks, or other properties or services of any third party including, but not limited to, Google or any mobile communications carrier.

4.5 You agree that you are solely responsible for (and that Google has no responsibility to you or to any third party for) any data, content, or resources that you create, transmit or display through Android and/or applications for Android, and for the consequences of your actions (including any loss or damage which Google may suffer) by doing so.

4.6 You agree that you are solely responsible for (and that Google has no responsibility to you or to any third party for) any breach of your obligations under the License Agreement, any applicable third party contract or Terms of Service, or any applicable law or regulation, and for the consequences (including any loss or damage which Google or any third party may suffer) of any such breach.

### 5. Your Developer Credentials

5.1 You agree that you are responsible for maintaining the confidentiality of any developer credentials that may be issued to you by Google or which you may choose yourself and that you will be solely responsible for all applications that are developed under your developer credentials.

### 6. Privacy and Information

6.1 In order to continually innovate and improve the SDK, Google may collect certain usage statistics from the software including but not limited to a unique identifier, associated IP address, version number of the software, and information on which tools and/or services in the SDK are being used and how they are being used. Before any of this information is collected, the SDK will notify you and seek your consent. If you withhold consent, the information will not be collected.

6.2 The data collected is examined in the aggregate to improve the SDK and is maintained in accordance with Google's Privacy Policy, which is located at the following URL: <https://policies.google.com/privacy>

6.3 Anonymized and aggregated sets of the data may be shared with Google partners to improve the SDK.

### 7. Third Party Applications

7.1 If you use the SDK to run applications developed by a third party or that access data, content or resources provided by a third party, you agree that Google is not responsible for those applications, data, content, or resources. You understand that all data, content or resources which you may access through such third party applications are the sole responsibility of the person from which they originated and that Google is not liable for any loss or damage that you may experience as a result of the use or access of any of those third party applications, data, content, or resources.

7.2 You should be aware the data, content, and resources presented to you through such a third party application may be protected by intellectual property rights which are owned by the providers (or by other persons or companies on their behalf). You may not modify, rent, lease, loan, sell, distribute or create derivative works based on these data, content, or resources (either in whole or in part) unless you have been specifically given permission to do so by the relevant owners.

7.3 You acknowledge that your use of such third party applications, data, content, or resources may be subject to separate terms between you and the relevant third party. In that case, the License Agreement does not affect your legal relationship with these third parties.

### 8. Using Android APIs

8.1 Google Data APIs

8.1.1 If you use any API to retrieve data from Google, you acknowledge that the data may be protected by intellectual property rights which are owned by Google or those parties that provide the data (or by other persons or companies on their behalf). Your use of any such API may be subject to additional Terms of Service. You may not modify, rent, lease, loan, sell, distribute or create derivative works based on this data (either in whole or in part) unless allowed by the relevant Terms of Service.

8.1.2 If you use any API to retrieve a user's data from Google, you acknowledge and agree that you shall retrieve data only with the user's explicit consent and only when, and for the limited purposes for which, the user has given you permission to do so. If you use the Android Recognition Service API, documented at the following URL: <https://developer.android.com/reference/android/speech/RecognitionService>, as updated from time to time, you acknowledge that the use of the API is subject to the Data Processing Addendum for Products where Google is a Data Processor, which is located at the following URL: <https://privacy.google.com/businesses/gdprprocessorterms/>, as updated from time to time. By clicking to accept, you hereby agree to the terms of the Data Processing Addendum for Products where Google is a Data Processor.

### 9. Terminating this License Agreement

9.1 The License Agreement will continue to apply until terminated by either you or Google as set out below.

9.2 If you want to terminate the License Agreement, you may do so by ceasing your use of the SDK and any relevant developer credentials.

9.3 Google may at any time, terminate the License Agreement with you if:

(A) you have breached any provision of the License Agreement; or

(B) Google is required to do so by law; or

(C) the partner with whom Google offered certain parts of SDK (such as APIs) to you has terminated its relationship with Google or ceased to offer certain parts of the SDK to you; or

(D) Google decides to no longer provide the SDK or certain parts of the SDK to users in the country in which you are resident or from which you use the service, or the provision of the SDK or certain SDK services to you by Google is, in Google's sole discretion, no longer commercially viable.

9.4 When the License Agreement comes to an end, all of the legal rights, obligations and liabilities that you and Google have benefited from, been subject to (or which have accrued over time whilst the License Agreement has been in force) or which are expressed to continue indefinitely, shall be unaffected by this cessation, and the provisions of paragraph 14.7 shall continue to apply to such rights, obligations and liabilities indefinitely.

.

### 10. Indemnification

10.1 To the maximum extent permitted by law, you agree to defend, indemnify and hold harmless Google, its affiliates and their respective directors, officers, employees and agents from and against any and all claims, actions, suits or proceedings, as well as any and all losses, liabilities, damages, costs and expenses (including reasonable attorneys fees) arising out of or accruing from

(a) your use of the SDK,

(b) any application you develop on the SDK that infringes any copyright, trademark, trade secret, trade dress, patent or other intellectual property right of any person or defames any person or violates their rights of publicity or privacy, and

(c) any non-compliance by you with the License Agreement.

### 11. Changes to the License Agreement

11.1 Google may make changes to the License Agreement as it distributes new versions of the SDK. When these changes are made, Google will make a new version of the License Agreement available on the website where the SDK is made available.

### 12. General Legal Terms

12.1 The License Agreement constitutes the whole legal agreement between you and Google and governs your use of the SDK (excluding any services which Google may provide to you under a separate written agreement), and completely replaces any prior agreements between you and Google in relation to the SDK.

12.2 You agree that if Google does not exercise or enforce any legal right or remedy which is contained in the License Agreement (or which Google has the benefit of under any applicable law), this will not be taken to be a formal waiver of Google's rights and that those rights or remedies will still be available to Google.

12.3 If any court of law, having the jurisdiction to decide on this matter, rules that any provision of the License Agreement is invalid, then that provision will be removed from the License Agreement without affecting the rest of the License Agreement. The remaining provisions of the License Agreement will continue to be valid and enforceable. 12.4 You acknowledge and agree that each member of the group of companies of which Google is the parent shall be third party beneficiaries to the License Agreement and that such other companies shall be entitled to directly enforce, and rely upon, any provision of the License Agreement that confers a benefit on (or rights in favor of) them.

.

**19. Latest version :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **SDK / API level** | [**Version code**](https://developer.android.com/reference/kotlin/android/os/Build.VERSION_CODES) | **Codename** | **Cumulative usage**[**1**](https://apilevels.com/#fn:1) | **Year** |
| ***Android 13******DEV*** | *Level 33* | *T* | *Tiramisu*[*2*](https://apilevels.com/#fn:2) | *No data* | *TBD* |
| [**Android 12**](https://developer.android.com/about/versions/12) | Level 32 **Android 12L** **BETA** | S\_V2 | Snow Cone [2](https://apilevels.com/#fn:2) | *No data* | TBD |
| Level 31 **Android 12** | S | 5.5% | 2021 |
| [**Android 11**](https://developer.android.com/about/versions/11) | Level 30 | R | Red Velvet Cake [2](https://apilevels.com/#fn:2) | 43.0% | 2020 |
| * **targetSdk** [must be 30+](https://developer.android.com/distribute/play-policies) for new apps and app updates. | | | |
| [**Android 10**](https://developer.android.com/about/versions/10) | Level 29 | Q | Quince Tart [2](https://apilevels.com/#fn:2) | 67.7% | 2019 |
| [**Android 9**](https://developer.android.com/about/versions/pie) | Level 28 | P | Pie | 79.9% | 2018 |
| [**Android 8**](https://developer.android.com/about/versions/oreo) | Level 27 **Android 8.1** | O\_MR1 | Oreo | 86.3% | 2017 |
| Level 26 **Android 8.0** | O | 89.1% |
| [**Android 7**](https://developer.android.com/about/versions/nougat) | Level 25 **Android 7.1** | N\_MR1 | Nougat | 90.9% | 2016 |
| Level 24 **Android 7.0** | N | 93.7% |
| [**Android 6**](https://developer.android.com/about/versions/marshmallow) | Level 23 | M | Marshmallow | 96.6% | 2015 |
| [**Android 5**](https://developer.android.com/about/versions/lollipop) | Level 22 **Android 5.1** | LOLLIPOP\_MR1 | Lollipop | 98.4% | 2015 |
| Level 21 **Android 5.0** | LOLLIPOP, L | *No data* | 2014 |
| * [Jetpack Compose](https://developer.android.com/jetpack/compose) requires a **minSdk** of 21 or higher. | | |
| **Android 4** | Level 19 [3](https://apilevels.com/#fn:3) **Android 4.4** | KITKAT | KitKat | 2013 |
| * Google Play services [do not support Android versions](https://android-developers.googleblog.com/2021/07/google-play-services-discontinuing-jelly-bean.html) below API level 19. | | |
| Level 18 **Android 4.3** | JELLYBEAN\_MR2 | Jelly Bean |
| Level 17 **Android 4.2** | JELLYBEAN\_MR1 | 2012 |
| Level 16 **Android 4.1** | JELLYBEAN |
| Level 15 **Android 4.0.3 – 4.0.4** | ICE\_CREAM\_SANDWICH\_MR1 | Ice Cream Sandwich | 2011 |
| Level 14 **Android 4.0.1 – 4.0.2** | ICE\_CREAM\_SANDWICH |
| **Android 3** | Level 13 **Android 3.2** | HONEYCOMB\_MR2 | Honeycomb |
| Level 12 **Android 3.1** | HONEYCOMB\_MR1 |
| Level 11 **Android 3.0** | HONEYCOMB |
| **Android 2** | Level 10 **Android 2.3.3 – 2.3.7** | GINGERBREAD\_MR1 | Gingerbread |
| Level 9 **Android 2.3.0 – 2.3.2** | GINGERBREAD | 2010 |
| Level 8 **Android 2.2** | FROYO | Froyo |
| Level 7 **Android 2.1** | ECLAIR\_MR1 | Éclair |
| Level 6 **Android 2.0.1** | ECLAIR\_0\_1 | 2009 |
| Level 5 **Android 2.0** | ECLAIR |
| **Android 1** | Level 4 **Android 1.6** | DONUT | Donut |
| Level 3 **Android 1.5** | CUPCAKE | Cupcake |
| Level 2 **Android 1.1** | BASE\_1\_1 | Petit Four |
| Level 1 **Android 1.0** | BASE | *None* | 2008 |

**20. Cloud support (Yes/No) :**

By default, Google Cloud blocks the ability to create nested virtual machines, so Android Studio runs, but you can't run an AVD using the emulator. ... For details of how nested virtualization works and what restrictions exist for nested virtualization, see Enabling nested virtualization for VM instances.

**21. Applicability :**

* The Android SDK (Software Development Kit) is a set of development tools that are used to develop applications for the Android platform.
* This SDK provides a selection of tools that are required to build Android applications and ensures the process goes as smoothly as possible.

* Whether you create an application using *Java,* Kotlin or *C#* you need the SDK to get it to run on any Android device. You can also use an emulator in order to test the applications that you have built.

**22. Drawbacks (if any) :**

Android users spending habits

Numerous research will tell you that Apple users spend way more on their apps than Android enthusiasts. Statista claims that App Store takes more than 60 percent of all mobile spendings while Google Play lags behind with a little over 30 percent.

Not only does it concern app downloads, but also in-app purchases. The reasons for that may be one or several of the following:

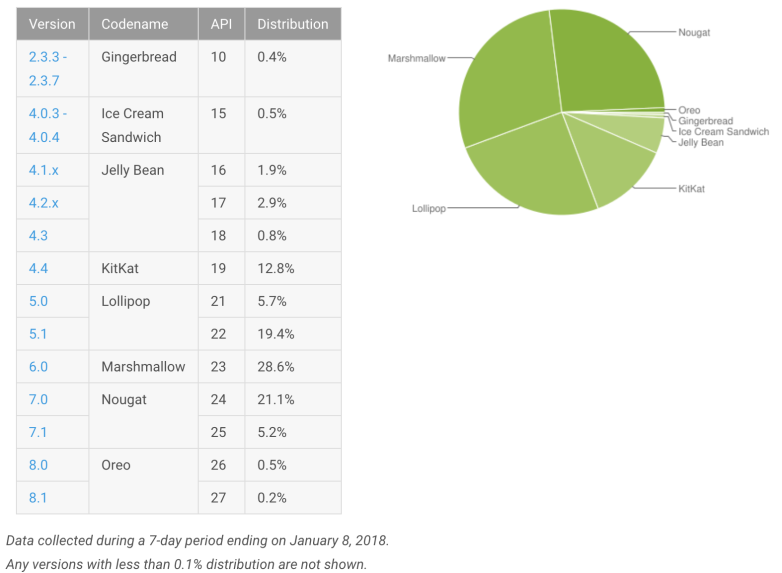
* The overpowering number of Google Play apps leads to a long list of alternatives for paid or freemium programs. Users simply can find more free apps similar to the paid ones
* Android and iOS share different demographics. Most Apple fans are located in North America, Australia, and Northern Europe, while Android is popular in lower-income countries
* During its long acceptance process, Apple weeds out the apps with the poor user experience. Therefore, more quality apps reach App Store top charts inducing users to pay up
* Many Android apps can be prone to bugs due to the big number of devices and active users of older operating systems on the market. Developers have to spend more time to thoroughly test apps on all models, which is often impossible. We will talk about it further.

Security issues

Although Android’s open-source nature is a blessing for developers, it can also be a curse. While malware and hacks target millions of Android users almost weekly, Google quickly releases security patches. Unfortunately, most people fail to update their phones regularly. This means that app developers often have to take care of user data themselves by conducting complex encryption, including additional security features, or avoiding personal data input altogether.

OS adoption fragmentation

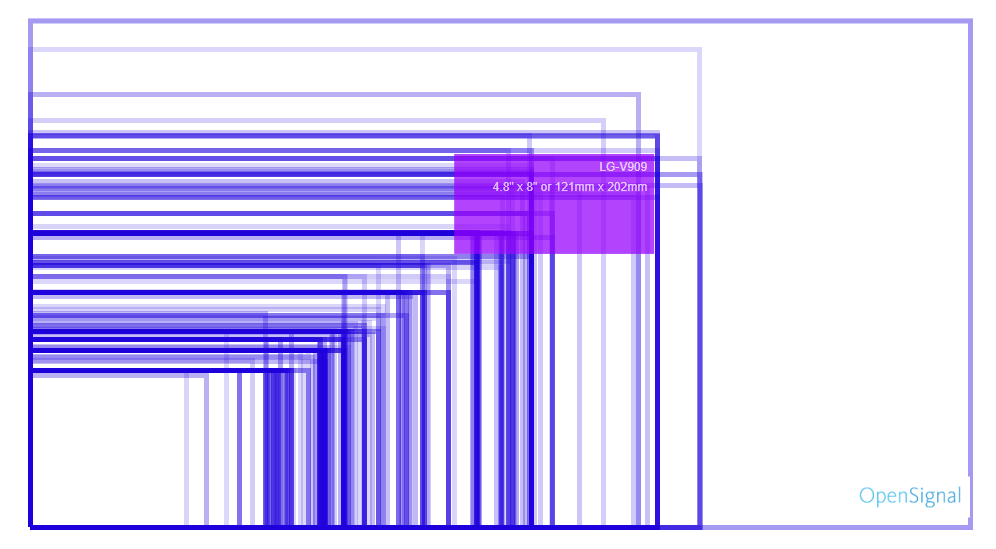
Testing for Android is a big, fat con. The range of operating systems is large and one of the challenges related to app customization. According to Google’s own [2018 report](https://developer.android.com/about/dashboards/), most Android devices are still running on Lollipop, Marshmallow, and Nougat  – operating systems from 2014, 2015, and 2016 respectively.



Another differentiation problem comes from various OEMs (original equipment manufacturers) that create their own “skins” making devices running on the same Android version look so different. Not only does it present another update-staggering layer, but also leads to changes in functionality that may affect how your app works on all smartphones. Sony and Samsung are the biggest offenders known to create complications for programmers.

Device fragmentation

The number of mobile manufacturers may be big but it’s still manageable compared to the number of devices these providers release – each with their own screen size, sensors, performance issues, and graphics drivers. In the official documentation for Material Design, [Google shares](https://material.io/tools/devices/) a long list of Android (and iOS) devices with corresponding screen dimensions and resolutions. Spoiler: It’s much bigger than Apple’s line of 14 devices.



Product owners have two solutions in the device fragmentation issue: Either limiting the number of supported devices and systems to a manageable maximum or lowering the app quality when advanced sensors or other functionality isn’t supported by older devices. This, however, doesn’t make development and testing efforts easy.

Copyright problems

Earlier we made a point that Google takes less time checking and accepting apps to Google Play, which means that you can release your MVP way easier compared to App Store. However, there’s another side to this advantage. First, Google doesn’t conduct thorough patent or copyright checks which means that you can unknowingly post a program that includes features or content already used in another product. Second, you can suffer from copyright infringement yourself when Google Play releases an app that repeats your unique solutions.