Contents

UNIT 1 - Getting started with Android

CHAPTER 1 - Introduction to Android

Structure of Introduction to Android

1.1 Learning Outcomes:

After the successful completion of this chapter, the student will be able to:

- Explain Android Operating System
- Differentiate between various mobile operating systems and the language used for android application development
- Discuss the history, versions, features of Android, and architecture of android
- Explain the components of the android application

1.2 Introduction to Operating System

An Operating System is an essential part of the computer system. It consists of a set of programs that are executed when the machine is turned on and manages the resources efficiently depending on the user's request. In other words, Operating System is system software that manages the computer's hardware and provides an environment for running programs. An Operating System is loaded into memory when the computer is booted and remains active as long as the machine is on.

It acts as a resource manager. The resources of a computer system include CPU, primary memory, secondary memory, input/output devices, files, etc. It keeps track of the status of each resource and decides who will have control over the computer resource for how long and when. Thus, the operating system becomes an interface between the user and the machine.

The tasks of an operating system are:

- Primary Memory Management
- Secondary Memory Management
- I/O System Management
- File Management
- Process Management

Primary Memory Management -

It is the functionality of an operating system that manages and handles the competing processes running in the main memory. It decides which process should get the memory, when and for how long. During execution, it moves the processes back and forth between the main memory and the disk. It also keeps track of every memory location irrespective of whether the memory location is allocated or unallocated.

Secondary Memory Management -

Secondary memory is non-volatile, which means it stores data even if the computer is turned off or until the data is deleted or overwritten. Since primary memory is not sufficient to store data and is volatile all programs are usually stored on disk. The operating system keeps track of the available space on the disk and allocates it properly to the programs being stored on it. The activities of secondary memory management are storage allocation, disk scheduling, and free-space management.

I/O System Management –

The task of this module is to keep track of the status of different devices connected to the computer and allocate the sharable devices properly. This process is usually done by the IO controller and IO scheduler.

File Management –

File management is the process of managing the files in the computer system. It includes the process of creating, updating, and deleting files and directories in the computer system.

Process Management -

A program is called a process during its execution. It is the task of the operating system to manage all running processes and allocate the CPU and resources as and when required.

1.3 Overview of Mobile Operating System

The Mobile operating system is a set of programs (software) used by small hand-held devices like smartphones, tablets, etc to run other applications. It manages all the hardware resources and controls their use. It also provides a platform for managing application programs and files.

The most popular operating systems used for smartphones, tablets, and other hand-held devices are

- Android OS – It is open-source software developed by Google
- Apple iOS – It is developed by Apple Inc and can be used only on Apple devices
- BlackBerry OS – It is developed by Research in Motion (RIM), especially for handheld devices of BlackBerry
- Windows Phone 7 – It is developed by Microsoft for smartphones and pocket PCs
- WebOS – It is developed by Palm and is based on Linux Kernel, used by HP for touchpads and smartphones

1.4 Android Operating System

1.4.1 What is Android

Android is an open-source Linux-based operating system used for mobile devices such as smartphones, tablets, electronic book readers, etc. It is currently used even in televisions and set-up boxes.

1.4.2 History and Versions of Android

Android development was started in 2003 by Android Inc., California. Andy Rubin, Rich Miner, Chris White, and Nick Sears were its founders. Later it was bought by Google Inc. in 2005. Android is continually been developed by the Open Handset Alliance, led by Google, and other companies. Various versions of Android are developed since 2009. The following figure and table show the various versions of Android. The code names of android range alphabetically from A to P, Android 10, Android 11, and Android 12 is the latest release in October 2021. Currently, Android 12 is the 19th version of Android and the 12th major release.

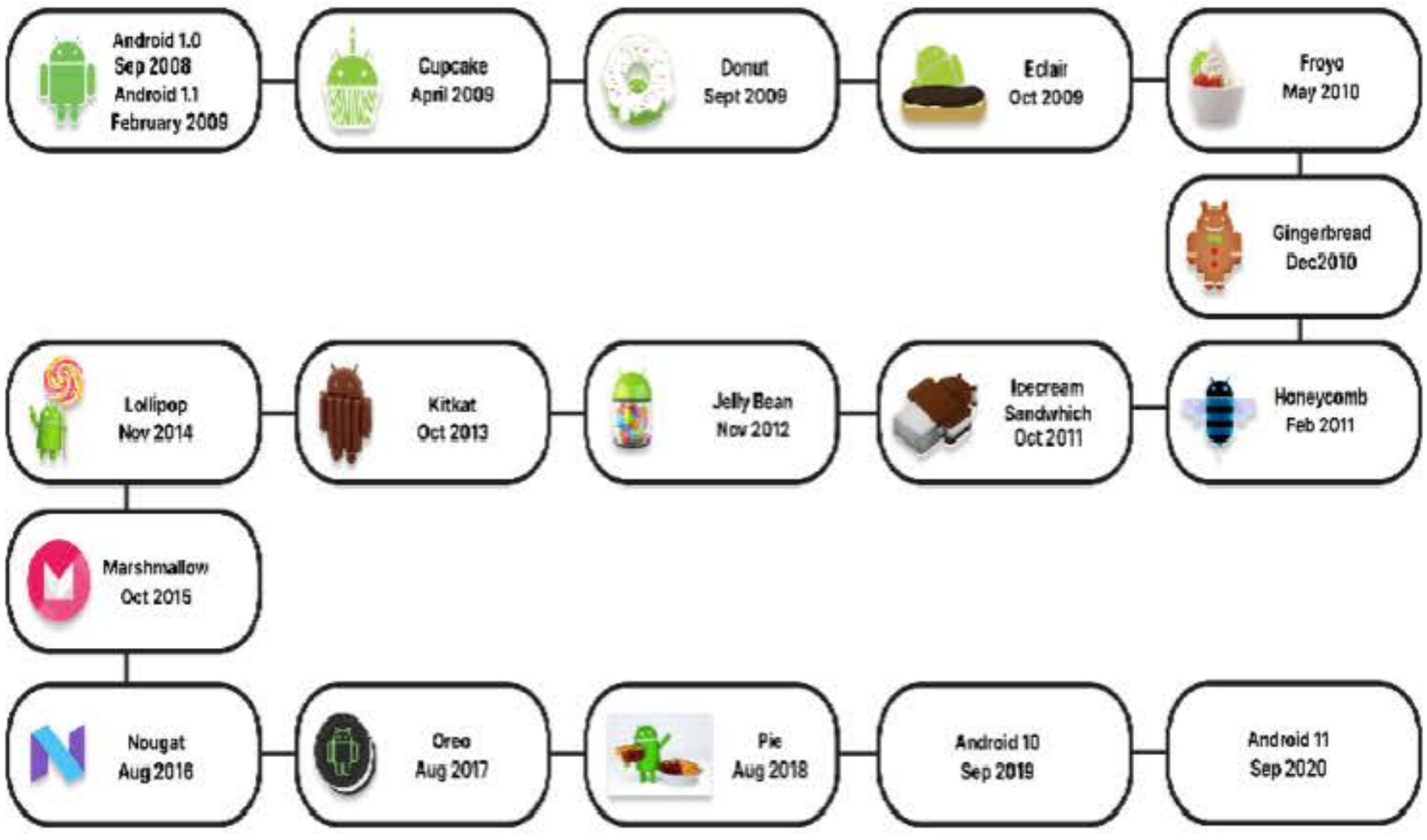

Figure 1.1: Various versions of Android with code names

Various versions of the Android platform are uniquely identified by exactly one API level which is an integer value. [Although backward compatibility is supported with lower API levels down to 1].

An application programming interface (API) level for Android consists of a set of packages, classes, methods, protocols, XML elements and attributes, tools, intents, and a set of permissions that can be used for building a software application and for interacting with the underlying Android system. An API acts as an interface for communicating between different software. Each successive version of Android includes updates i.e. addition of new or replacement of existing functionalities to the API that it delivers. Each API level has its functionalities with support for backward compatibility.

Android Studio provides two terminologies for setting minimum and maximum levels.

- targetSdkVersion – It is the version for which the application is being developed
- minSdkVersion - It is the minimum version that will support the application

Thus the application is compatible with all versions between targetSdkVersion down to minSdkVersion.

Table 1.1: API Levels and Code names of Android

Android Platform version	API Level	Version Code
Android 1.0	1	BASE
Android 1.1	2	BASE_1_1
Android 1.5	3	CUPCAKE
Android 1.6	4	DONUT
Android 2.0	5	ÉCLAIR
Android 2.0.1	6	ÉCLAIR_0_1
Android 2.1.x	7	ÉCLAIR_MR1
Android 2.2.x	8	FROYO
Android 2.3, 2.3.1, 2.3.2	9	GINGERBREAD
Android 2.3.3, 2.3.4	10	GINGERBREAD_MR1
Android 3.0.x	11	HONEYCOMB
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.2	13	HONEYCOMB_MR2
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.3	18	JELLY_BEAN_MR2
Android 4.4	19	KITKAT
Android 4.4W	20	KITKAT_WATCH
Android 5.1	22	LOLLIPOP_MR1
Android 6.0	23	M
Android 7.0	24	N
Android 7.1, 7.1.1	25	N_MR1
Android 8.0	26	O
Android 8.1	27	O_MR1
Android 9	28	P
Android 10	29	Q
Android 11	30	R
Android 12	31	S

1.4.3 Features of Android

- User Interface- Android provides a user-friendly user interface
- Messaging – It supports SMS and MMS
- Multitasking – Users can run various applications simultaneously at the same time. Some apps and services can be run in the background
- Multiple Language Support - It supports multiple languages like English, Germany, French, Japanese, Hindi, etc.
- Connectivity – It has extensive support for connectivity such as Wi-Fi, Bluetooth, GSM/EDGE, CDMA, VOLTE, VPN, EV-DO, Hotspot, NFC, WiMAX, 3G network band, 4G Network Band, etc.
- Multi-touch – It has support for multi-touch.
- Auto-Correction and Dictionary – It has an auto-correction and Dictionary facility.
- Storage – For storing data a lightweight relational database SQLite is used.
- Resizable widgets – It supports resizable widgets which can be expanded and shrunk to save space.

The latest versions of Android have various other features like audio device sharing, biometric access, limited storage access for apps, restricted access to cameras, and increased privacy.

1.4.4 Android Applications

The applications which are developed using Android are called “Android Apps”.

Different types of Android applications are

- Gaming Apps – Angry Birds, Subway Surfer, Halfbrick Studio games, Square Enix games, Pokemon Go, Subway Surfers, Supercell, etc.
- Educational Apps – Quizlet, Google Play books, Language learning apps, Flipboard, etc.
- Life Style Apps – Tripadvisor, OpenTable, Uber, food, etc.
- Social Media Apps – Facebook, Instagram, Snapchat, etc.
- Entertainment Apps – YouTube, Netflix, etc,
- News/Information outlets – Buzzfeed, Google News & weather, Smart news, etc.
- Utility Apps –Reminders, QR scanners, alarm clocks, calculators, etc are examples of utility apps that can be used by users for day-to-day activities.

1.5 Architecture of Android

The Architecture of Android consists of a stack of different software components distributed across four layers and five sections as shown in figure 1.2.

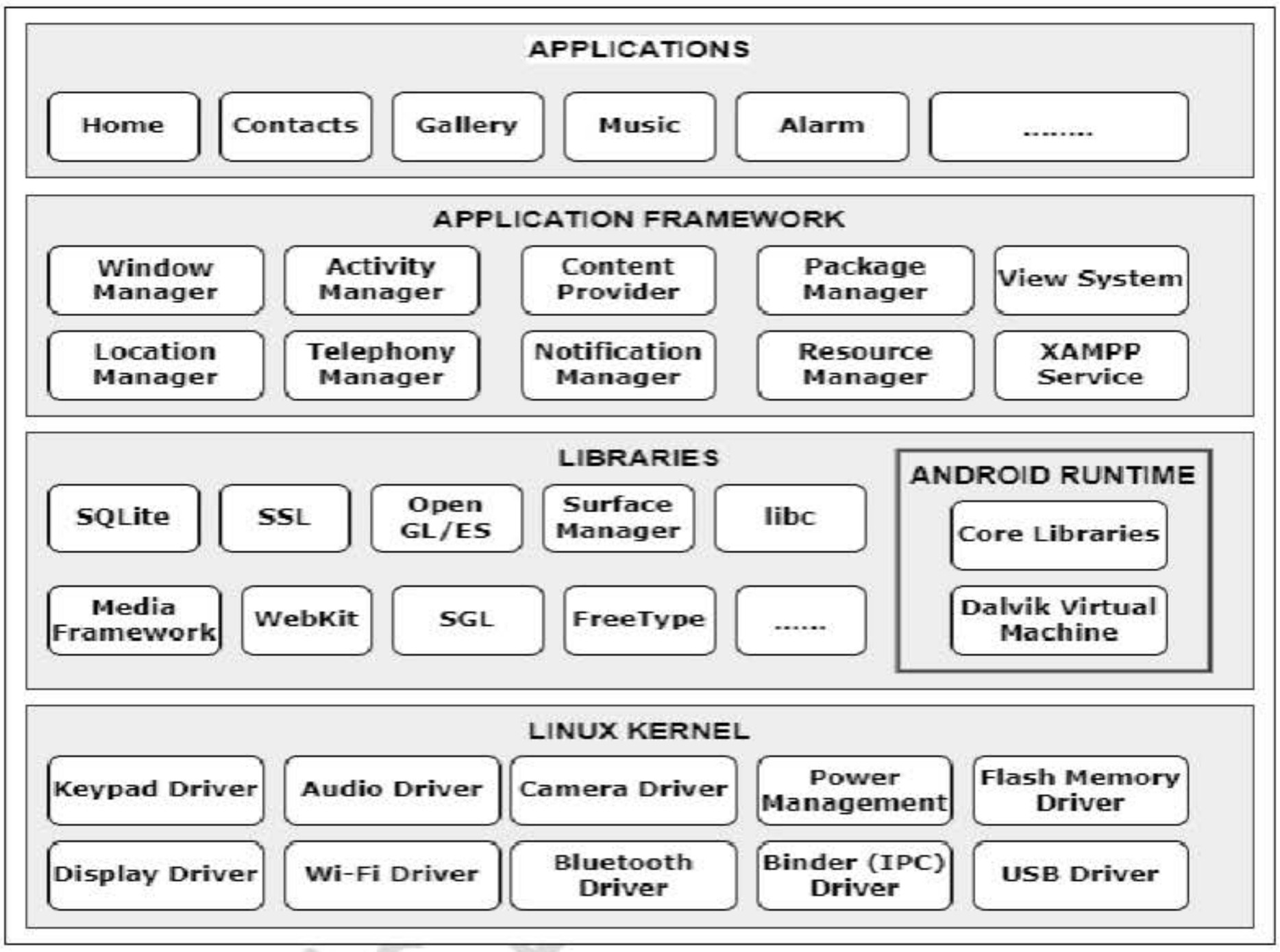

Figure 1.2: Architecture of Android

The main components of the android architecture are:

- Linux Kernel
- Native Libraries
- Android Runtime
- Android Application Framework
- Applications

Linux Kernel:

Linux Kernel is the lowest layer that is close to the hardware. The interaction between the software and hardware is provided by this layer. It serves as an abstraction layer for all other upper layers and consists of essential hardware drivers like keypad driver, display

driver, audio driver, Wi-fi driver, camera driver, memory driver, etc. The Linux kernel manages all the drivers which are required by the android device during runtime.
Linux Kernel is also responsible for power management, device management, memory management, and resource access.

Native Libraries:

On top of the Linux Kernel are the Native libraries like SQLite, Media, SSL, WebKit, OpenGL, 'C' runtime library (libc), etc. which provide support for Android development.
SQLite – is for database support and is used for storing and sharing application data.
Media – is for multimedia support. It is used to play and record audio and video.
SSL – is for Internet Security
WebKit – is for web browser support. It is used to render and display HTML (Hyper-Text Markup Language).
OpenGL and SGL – are used for rendering 2D and 3D graphics.

Android Runtime:

Android Runtime and Native Libraries are on the same layer just above the Linux kernel. It consists of core libraries and Dalvik Virtual Machine which is responsible to run an Android Application. Dalvik Virtual Machine usually called DVM is similar to JAVA Virtual Machine (JVM), but DVM is specially designed and used for small platform android devices which use low-power CPUs and less memory. DVM makes use of multithreading and memory management features like Linux. It uses less space and provides fast performance without affecting battery life.
The working of the Dalvik virtual machine along with the Dex Compiler is shown in figure 1.3.

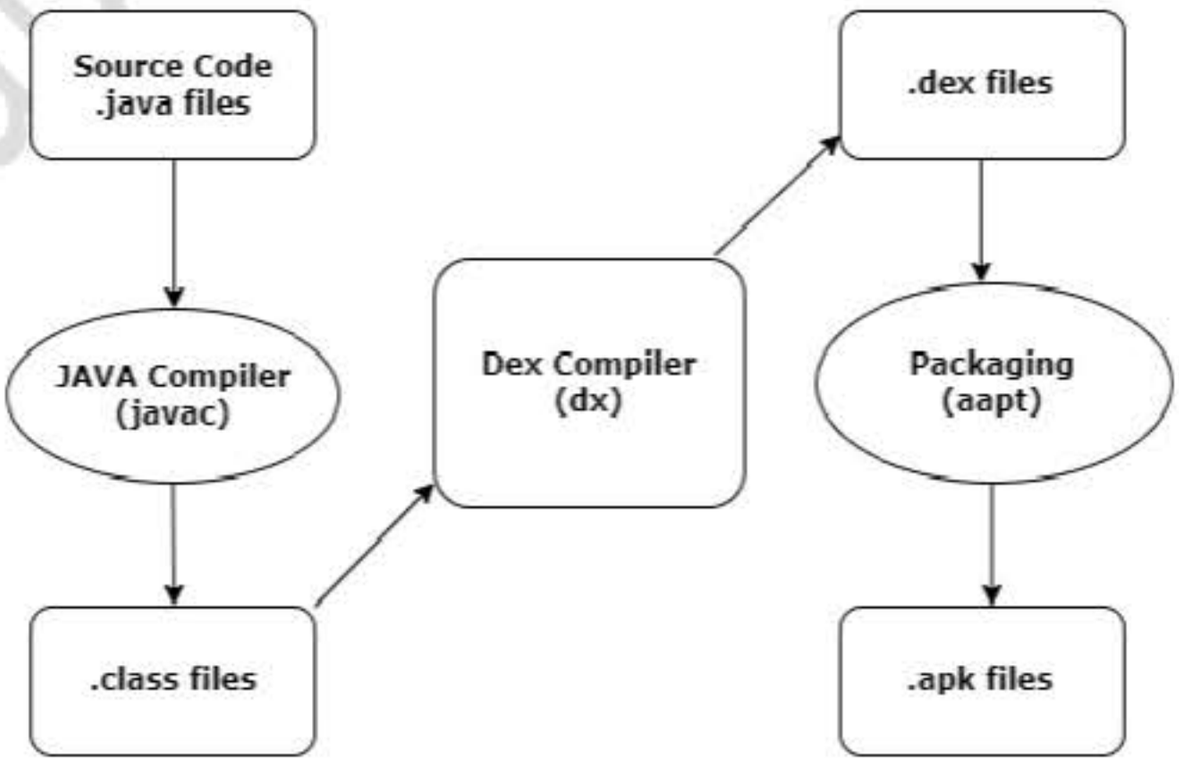

Figure 1.3: Working of Dex Compiler

The Source code (.java file) written in JAVA is converted by the JAVA compiler (javac) into a .class file. The Dex compiler then converts the generated .class file into a .dex file which in turn is later converted into a .apk file by the Android Asset Packaging Tool (AAPT).

Application Framework:

This layer resides above Native Libraries and Android Runtime. It provides a set of classes, interfaces, and resources used for android application development.

The basic building blocks of the Android Application framework are -

Activity Manager - It controls and manages the activity life cycle of an application.

Content Provider - It manages the publishing and sharing of data by applications.

Location Manager - It manages activities related to location management using GPS or a cell tower. It is the main class through which an application can access location services.

Telephony Manager - It provides various classes and interfaces which can be used to get network or telephony information and device details. It provides functionalities like calls, SMS, MMS, data services, etc.

Resource Manager – It organizes and manages resources like static content that the code uses such as bitmaps, strings, colors, user interface layouts, etc.

Application:

It is the topmost layer of android architecture with which the end-users interact. Pre-installed applications like gallery, camera, contacts, browsers, and third-party applications which are downloaded from the play store like games, chat applications, etc are all installed/deployed on this layer.

1.6 Components of the Android Application

The main core building blocks of android are activities, fragments, intents, views, services, content providers, and broadcast receivers. They are also called fundamental components. These components are used to develop Android Apps by application developers. The information about the components used in the application development and how they interact with each other is stored in the file named AndroidManifest.xml. This file is called a manifest file and it is the configuration file of an application.

1.6.1 Activity

An Activity is a single window screen that contains the User Interface of the application. An application may have zero or more activities. It is like an entry point for interacting with the user. Usually, a user can see a single screen called an activity at a time even though the

application may have multiple activities. The first screen that appears when an application is launched is known as “Main Activity”. A user can navigate back and forth among the activities.

Fragments:

Fragments are smaller activities that can be grouped to form an activity. An activity can display more than one fragment on the screen at a time. A part of an activity can be called a fragment. It represents a part of a user interface of an activity.

Intents:

Used to navigate between activities. It is used to invoke various components.

Views:

A view is the widget or user interface elements such as textbox, button, label, etc. Anything visible on a screen is a view. One or more views can be together grouped to form a view group.

1.6.2 Service

A Service does not have a User Interface. It is a process that runs in the background for a long time. For example, playing music in the background or fetching data from a database over a network without any user interaction.

1.6.3 Content Provider

It manages sharing of data by the applications based on permissions. It allows secured access and updates to data based on user requirements. Data used by the android applications can be stored in files, databases, or on the network. It acts as a mediator between the applications and data.

1.6.4 Broadcast Receivers

Broadcast Receivers are used to send or receive broadcast messages from the system or other applications. It follows a publish-subscribe pattern and is used for Asynchronous inter-process communication. The registered applications get a notification when an event occurs. Broadcast receivers do not have a User-Interface. The registered users receive notifications on the status bar. Some examples of broadcast receivers are low-battery, download complete notification, date changed, connectivity change, etc.

1.7 Self-Assessment Questions

Q1. What is Android? List and explain the features of Android. [8 marks, L2]

Q2. With a neat diagram explain the Architecture of Android.[8 marks, L2]

Q3. Discuss Android Runtime and Dalvik Virtual Machine. [8 marks, L2]

Q4. Explain in brief the basic building blocks of the Android Application framework. [6 marks, L2]

Q5. List and explain in brief the Components of the Android Application. [10 marks, L2]

Q6. Make a list of different types of android apps and name a few used by most of the People. [4 marks, L2]

Q7. Give an overview of an Android operating system.[6 marks, L2]

1.8 Self-Assessment Activities

A1. List the different operating systems used with smartphones and make a comparative study of them

A2. Discuss the advantages and disadvantages of the android operating system

1.9 Multiple-Choice Questions

1. Android Operating System uses _____. [1 mark, L1]
   a) Java Virtual Machine
   b) Dalvik Virtual Machine
   c) VirtualBox
   d) VMware Fusion

2. APK stands for _____. [1 mark, L1]
   a) Android Package Kit
   b) Android Platform Kit
   c) Application Package Kit
   d) None of the above

3. API stands for _____. [1 mark, L1]
   a) Application Programming Interface
   b) Android Programming Interface
   c) Android Packaging Interface
   d) None of the above

4. Java Byte code is converted into Dalvik Byte Code by _____. [1 mark, L1]
   a) Just In Time Compiler
   b) Dex Compiler
   c) Java Compiler

d) 'C' Compiler

5. The topmost layer of Android Architecture is _____ layer. [1 mark, L1]
   a) Native Libraries
   b) Application
   c) Linux Kernel
   d) Android Application Framework

6. The lowest layer of Android Architecture is _____ layer. [1 mark, L1]
   a) Native Libraries
   b) Application
   c) Linux Kernel
   d) Android Application Framework

7. Android is _____. [1 mark, L1]
   a) a web browser
   b) a web server
   c) a web client
   d) an operating system

8. Data sharing between applications is managed by _____. [1 mark, L1]
   a) Content Provider
   b) Telephony Manager
   c) Resource Manager
   d) Location Manager

9. AAPT in Android stands for _____. [1 mark, L1]
   a) Android Application Packaging Tool
   b) Advanced Android Packaging Tool
   c) Android Asset Packaging Tool
   d) Android Application Perfection Tool

10. An Android Operating System consists of ____ layers. [1 mark, L1]
    a) 3
    b) 4
    c) 5
    d) 2

1.10 Key Answers to Multiple-Choice Questions

1. Android Operating System uses Dalvik Virtual Machine.[b]
2. APK stands for Android Package Kit.[a]
3. API stands for Application Programming Interface.[a]
4. Java Byte code is converted into Dalvik Byte Code by Dex Compiler.[b]
5. The topmost layer of Android Architecture is the Application layer.[b]
6. The lowest layer of Android Architecture is the Linux Kernel layer.[c]
7. Android is an operating system.[d]
8. Data sharing between applications is managed by Content Provider.[a]
9. AAPT in Android stands for Android Asset Packaging Tool.[c]
10. An Android Operating System consists of 4 layers.[b]

1.11 Summary

Operating System is system software that manages the computer's hardware and provides an environment for running programs. An Operating System is loaded into memory when the computer is booted and remains active as long as the machine is on.

The tasks of an operating system are:

- Primary Memory Management
- Secondary Memory Management
- I/O System Management
- File Management
- Process Management

The Mobile operating system is a set of programs (software) used by small hand-held devices like smartphones, tablets, etc to run other applications. It manages all the hardware resources and controls their use. It also provides a platform for managing application programs and files.

Android is an open-source Linux-based operating system used for mobile devices such as smartphones, tablets, electronic book readers, etc. It is currently used even in televisions and set-up boxes.

The Architecture of Android consists of a stack of different software components distributed across four layers and five sections. The main components of the android architecture are: Linux Kernel, Native Libraries, Android Runtime, Android Application Framework, Applications

The main core building blocks of android are activities, fragments, intents, views, services, content providers, and broadcast receivers. They are also called

fundamental components. These components are used to develop Android Apps by application developers.

Activity: An Activity is a single window screen that contains the User Interface of the application.

Fragments: are smaller activities that can be grouped to form an activity.

Intents: Used to navigate between activities. It is used to invoke various components.

Views: A view is the widget or user interface elements such as textbox, button, label, etc. Anything visible on a screen is a view. One or more views can be together grouped to form a view group.

Service: A Service does not have a User Interface. It is a process that runs in the background for a long time.

Content Providers: It manages sharing of data by the applications based on permissions. It allows secured access and updates to data based on user requirements.

Broadcast Receivers: These are used to send or receive broadcast messages from the system or other applications. It follows a publish-subscribe pattern and is used for Asynchronous inter-process communication.

1.12 Keywords

- Android
- Dalvik Byte Code
- Java Byte Code
- Dex Compiler
- Content Provider
- Android Asset Packaging Tool (AAPT)

1.13 Recommended resources for further reading

Text Book(s)

1. Pradeep Kothari, "Android Application Development" Black-Book, Dreamtech Press,2015
2. Wei-Meng Lee, "Beginning Android Application Development", Wiley 2011.
3. Dawn Griffiths and David Griffiths, "Head First Android Development", O'Reilly, 2017.

References

- https://www.javatpoint.com/android-tutorial
- https://www.tutorialspoint.com/android/index.htm

- https://developer.android.com/guide/components/fundamentals

--*--

Contents

UNIT 1 – Getting started with Android

CHAPTER 2 - Configuring Android Development Environment

Structure of Configuring Android Development Environment

2.1 Learning Outcomes

After the successful completion of this chapter, the student will be able to:

- Install JAVA Development Kit (JDK), Android Studio, and the necessary tools
- Setup Android Virtual Device (AVD) for running an android application
- Demonstrate the creation of a basic android application

2.2 Downloading and Installing JDK and Android Studio

Depending on the operating system i.e. Windows, Linux, or mac OS before installing Android Studio the appropriate version of Java Development Kit (JDK) has to be installed as Java is the language used for developing an Android Application.

2.2.1 Installing JAVA Development Kit (JDK)

The latest version of JAVA 17 can be downloaded by using the following link
https://www.oracle.com/java/technologies/downloads/#jdk17-windows

Choose the appropriate operating system, when Windows Operating System is selected the following information is displayed

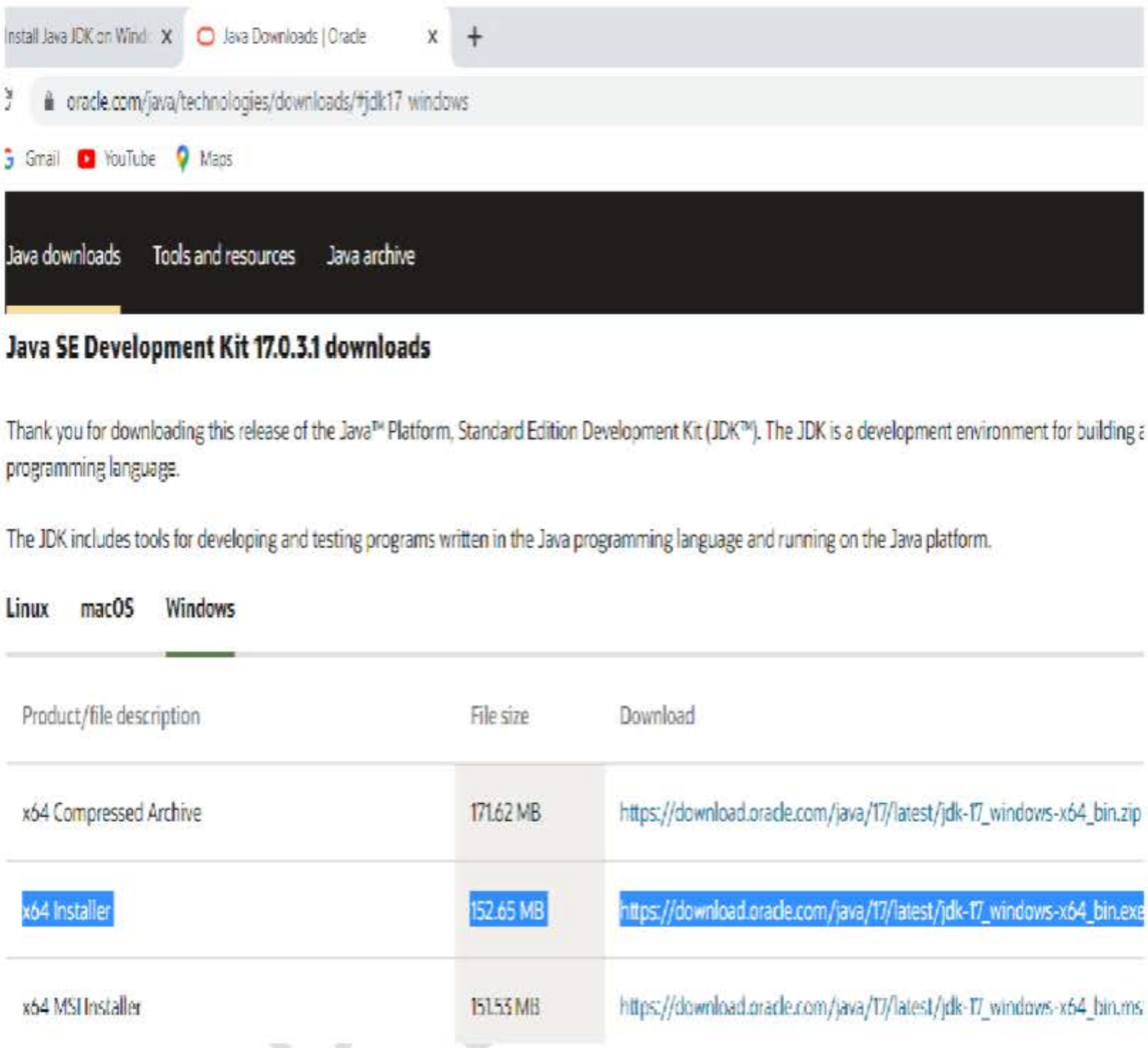

Figure 2.1: Downloading and Installing Java Development Kit (JDK)

Select download for x64 installer :

https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe

Once the file is downloaded double click on the downloaded file, set up JDK, and set the JAVA_HOME environment variable path appropriately.

2.2.2 Installing Android Studio

A minimum of 8GB RAM is required for running an android application in android studio using an android emulator, otherwise, the performance is very slow.

Android Studio and SDK tools can be downloaded by following the link

https://developer.android.com/studio

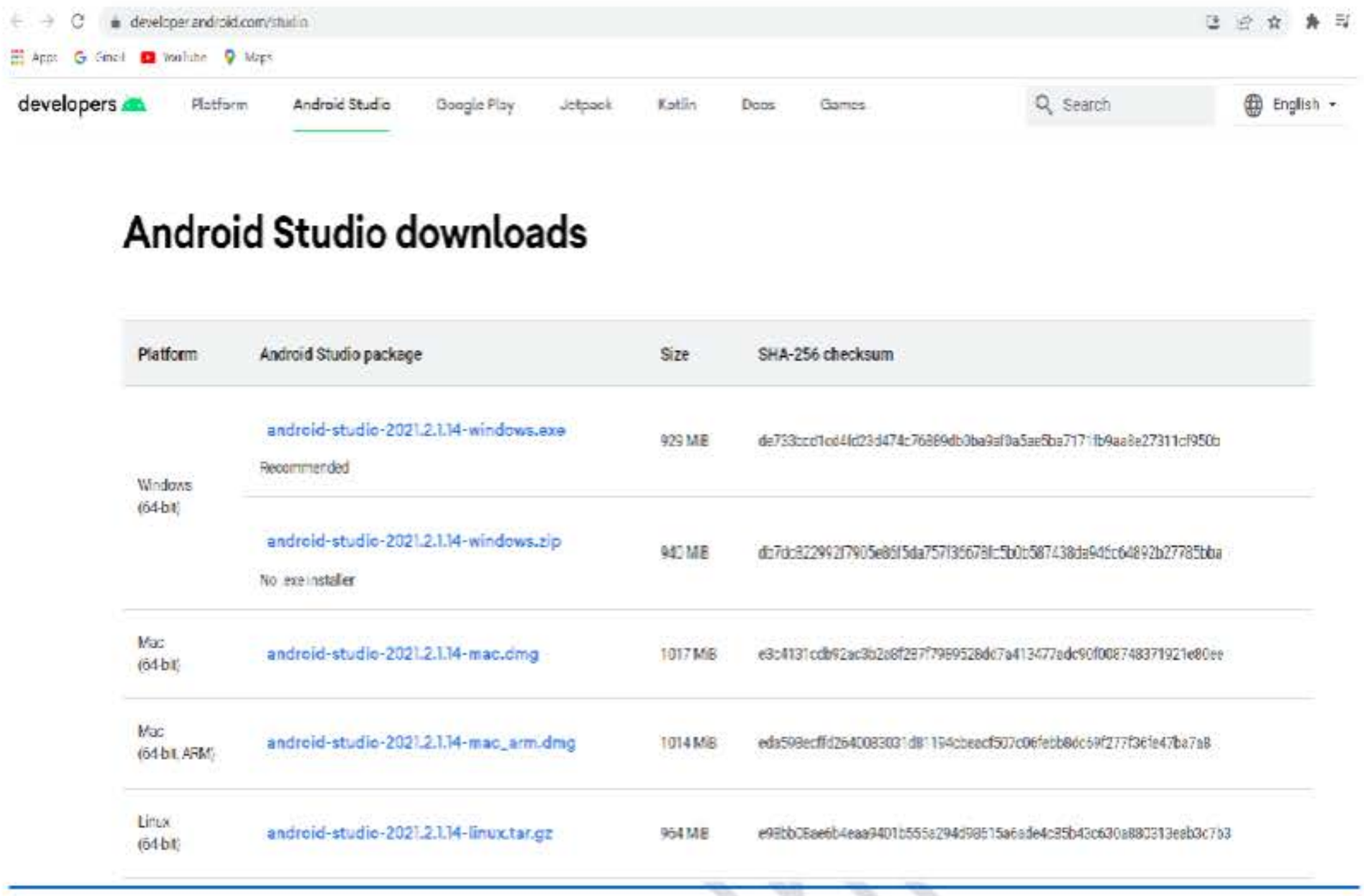

Figure 2.2: Downloading and Installing Android Studio

Select the appropriate platform based on the operating system used on the PC or laptop.
For Windows 10 (64-bit) download android-studio-2021.2.1.14-windows.exe and double-click on it to install android studio. It will take a few minutes, follow the setup wizard in android studio and finish the installation.
For downloading and installing Android Studio and for more information, follow the link to install Android Studio on the PC or laptop https://developer.android.com/ studio/install.
The video on the website shows each step of the setup procedure.

2.3 Setting up Android Virtual Device (AVD)

Android Virtual Device

An Android Virtual Device (AVD) is used for testing an Android Application. We can create any number of AVDs with different configurations to test android applications.
Each AVD has different configurations which consist of hardware profile, emulated storage, system image, API level, and other properties.
An Android Emulator is an AVD that represents a specific Android device. An Emulator runs the Android operating system in an AVD and provides all the capabilities of a real Android device. We can simulate text messages, calls, device location, etc on an emulator. Running android applications on emulators is faster than on a real device. Transfer of data is faster on an emulator.

To create an Android Virtual Device

1. In Android Studio from the Tools drop-down select Device Manager
2. On Opening the Device Manager window, click on Create device

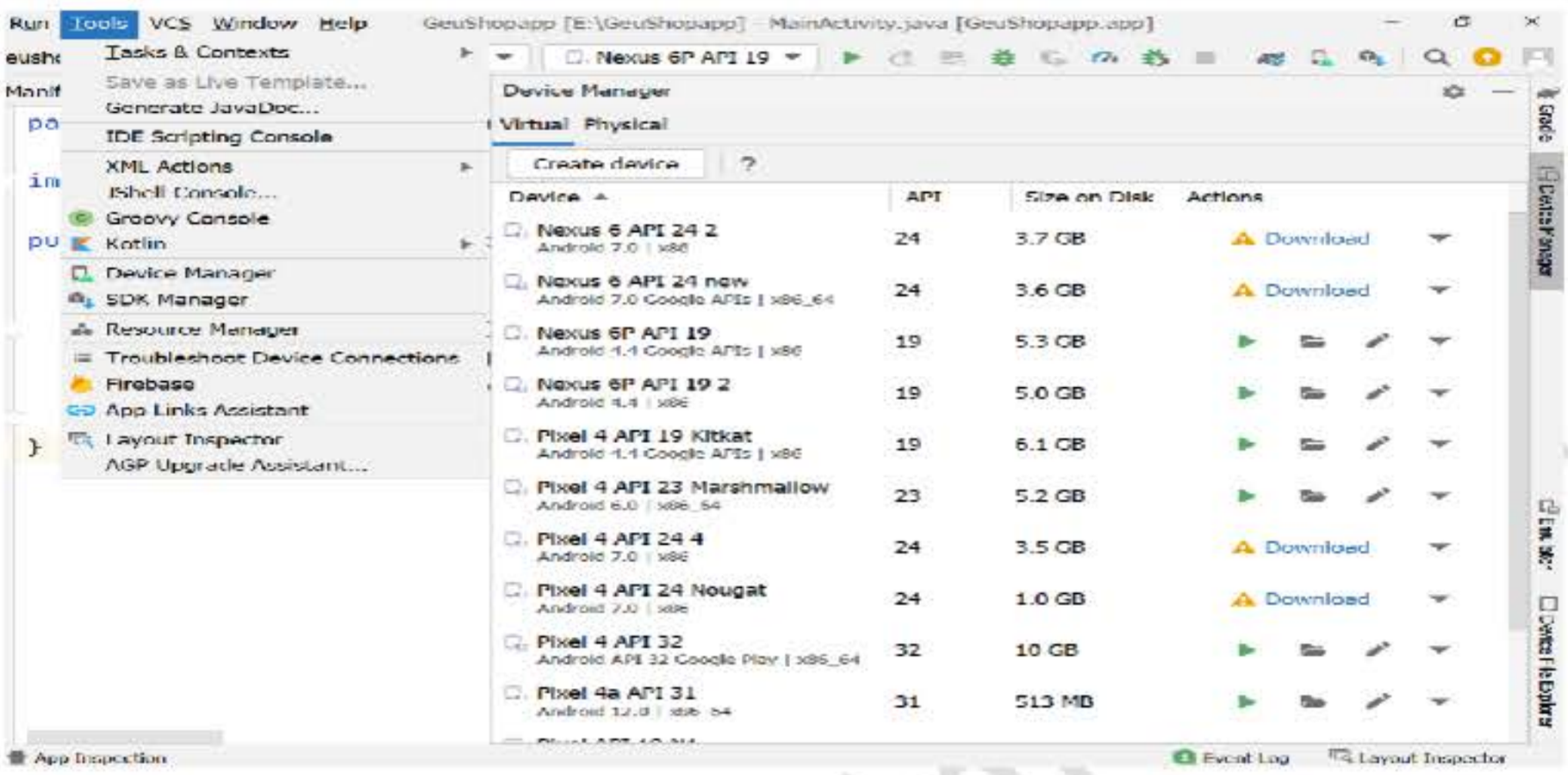

Figure 2.3: Device Manager in Android Studio

3. The Select Hardware window appears showing a list of hardware devices that are already configured. Choose a device such as Pixel and click Next

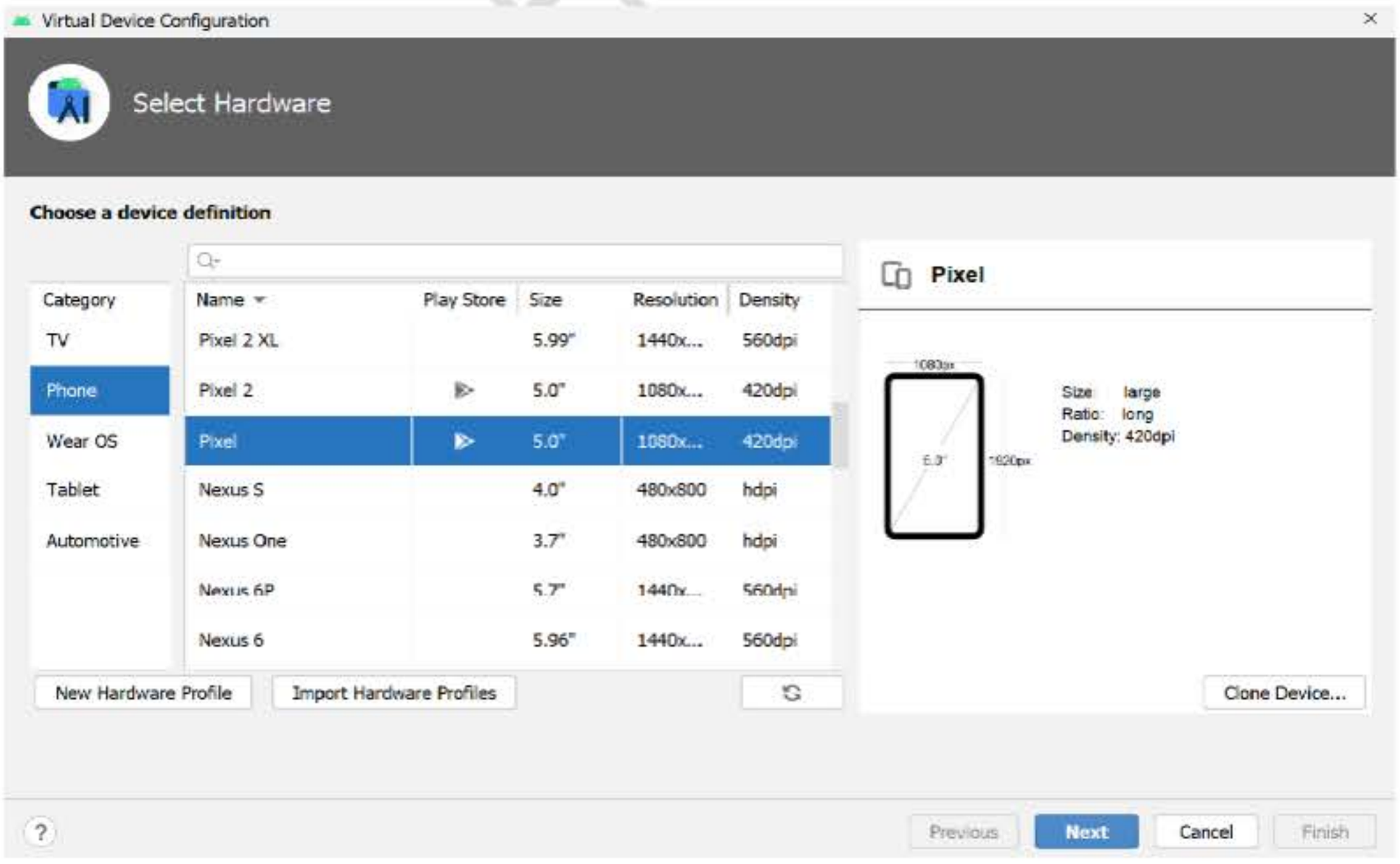

Figure 2.4: Select Hardware Window in Android Studio AVD

4. A system image window appears, from x86 images, select the API level of the virtual device (such as Kitkat) and start downloading if not downloaded and then click on Next

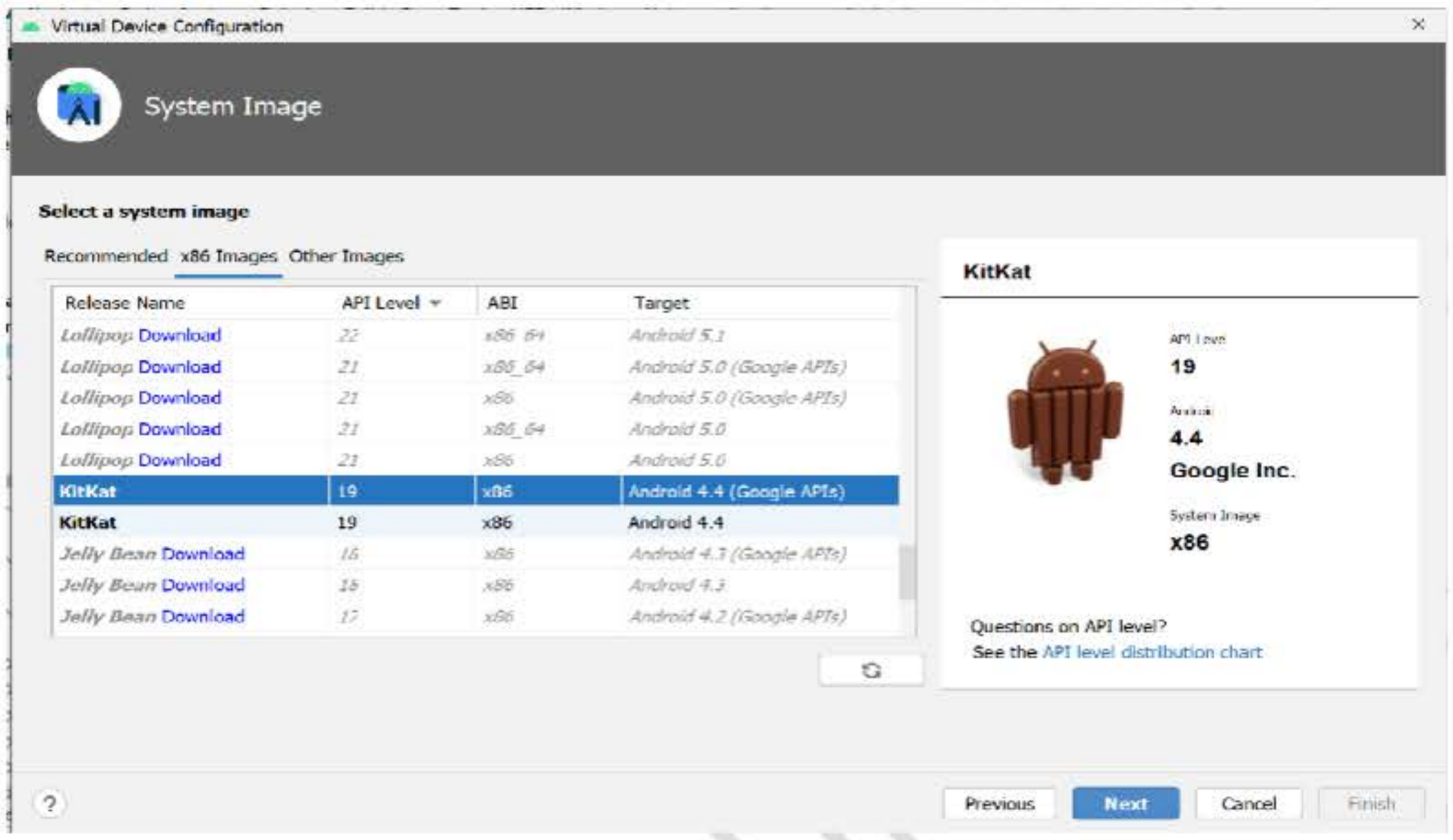

Figure 2.5: System Image in Android Studio

5. The Android Virtual Device (AVD) window appears, the AVD name can be changed, and then click on Finish. The Android Virtual Device by the name "**Pixel API 19 N4**" as given in the "verify configuration" is created.

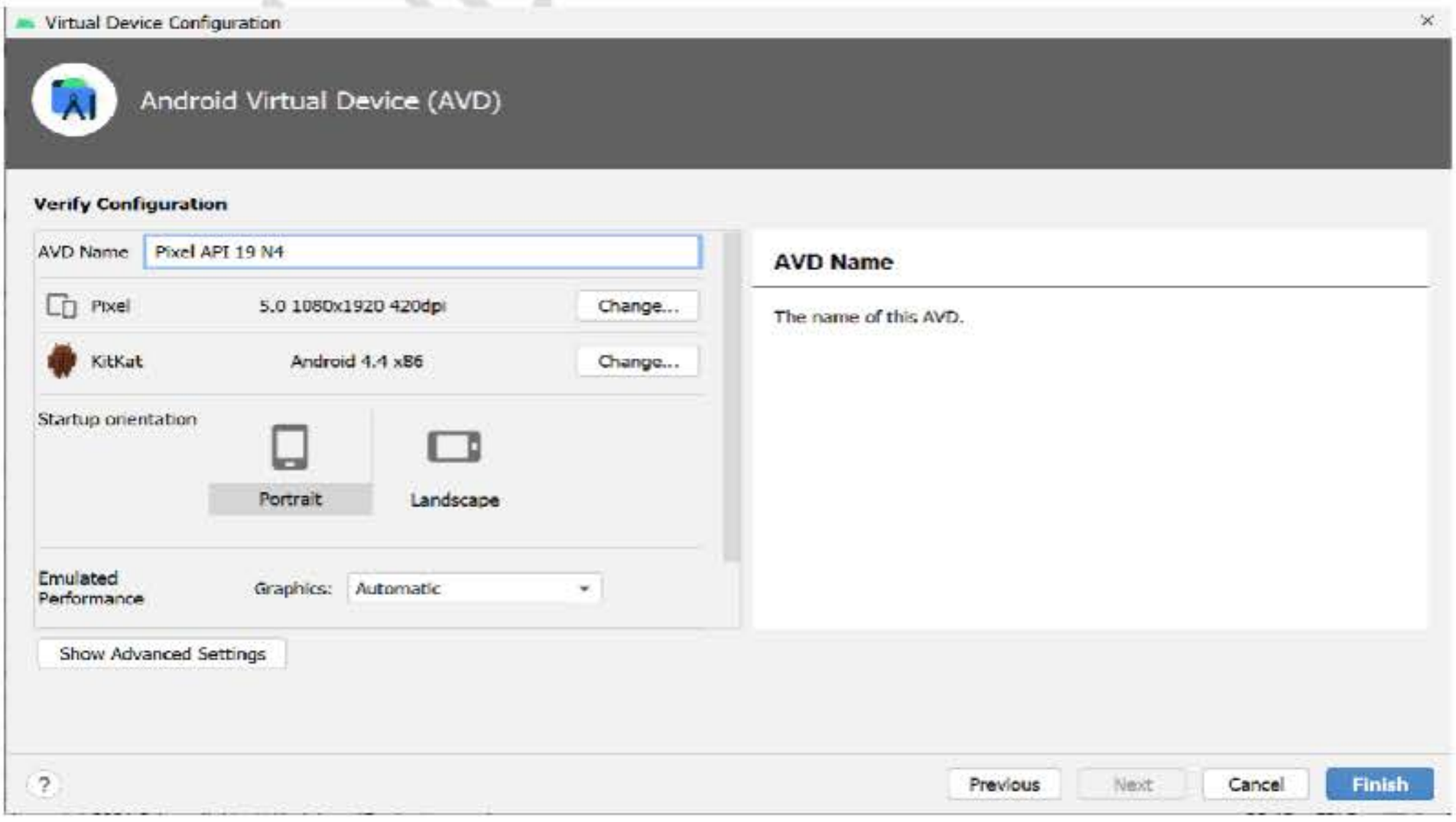

Figure 2.6: Android Virtual Device configuration in Android Studio

2.4 Creating First Android App: Creating a new Android Project

To create a new project in Android Studio follow the steps given below

1. Open Android Studio, click on New Project

   If a project is already open, select File -> New -> New Project

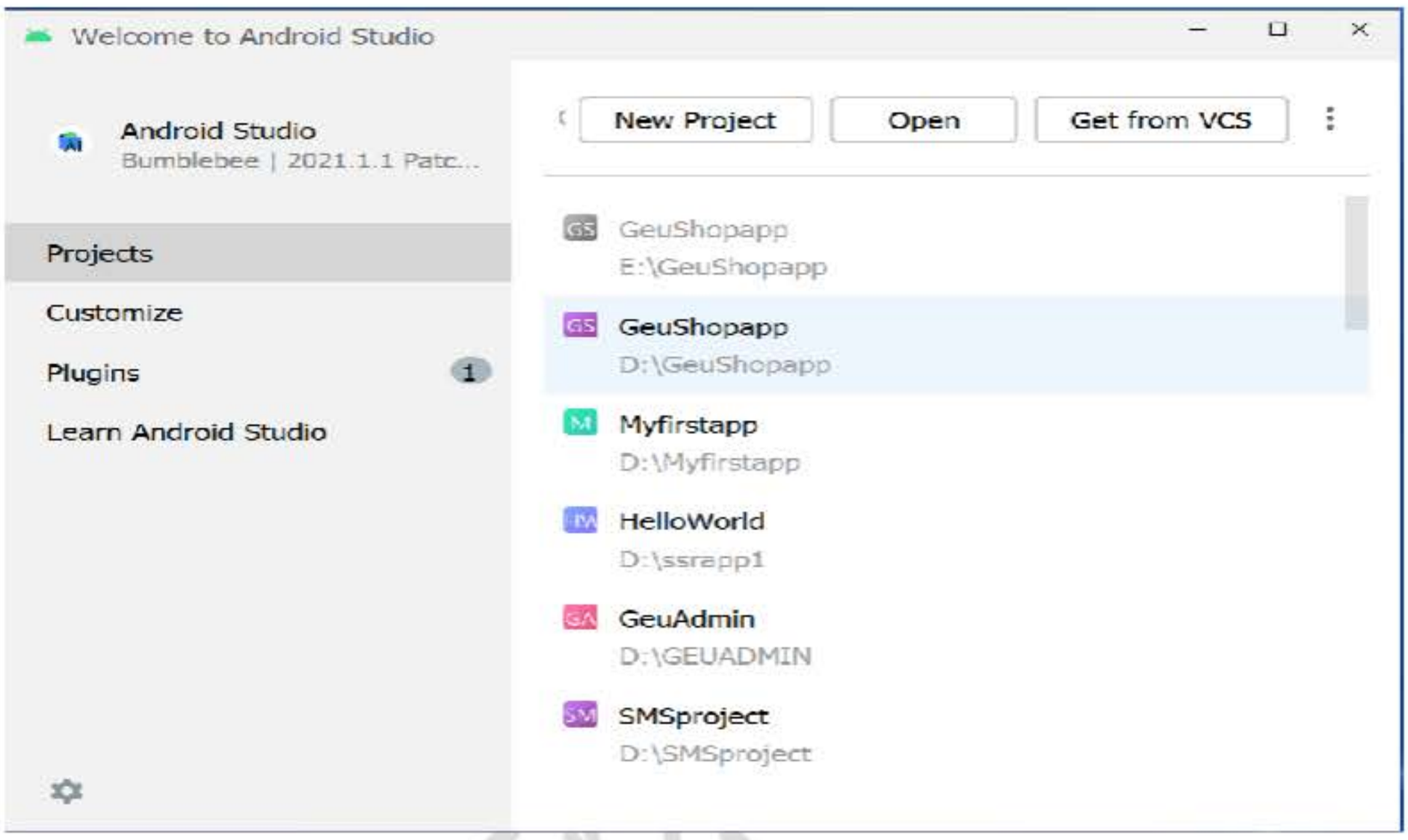

Figure 2.7: Android Studio New Project Window

2. In the New Project template, select Empty Activity and click Next

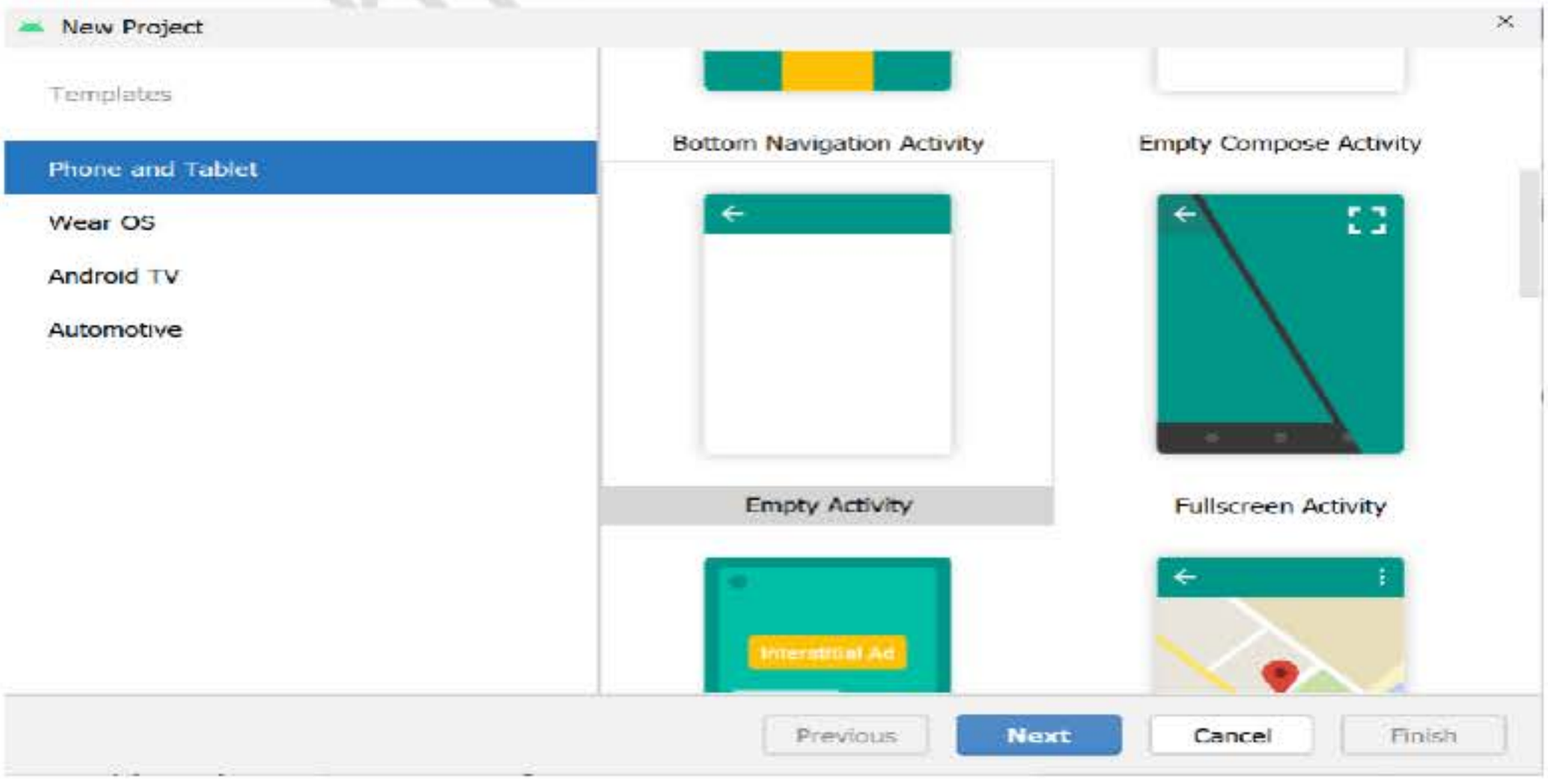

Figure 2.8: Android Studio Project template

3. In the Empty Activity Window, enter the following information
    1. Name of the Project: GeuShopapp
    2. Package Name: Can use the default - com.example.geushopapp package name or it can be changed
    3. Save location: Select the folder where the project has to be saved
    4. Language: Java is the default
    5. Minimum SDK: Select API 19: Android 4.4 (KitKat) from the drop-down so that the app will run on approximately 99.6% of devices
    6. Click on Finish Button

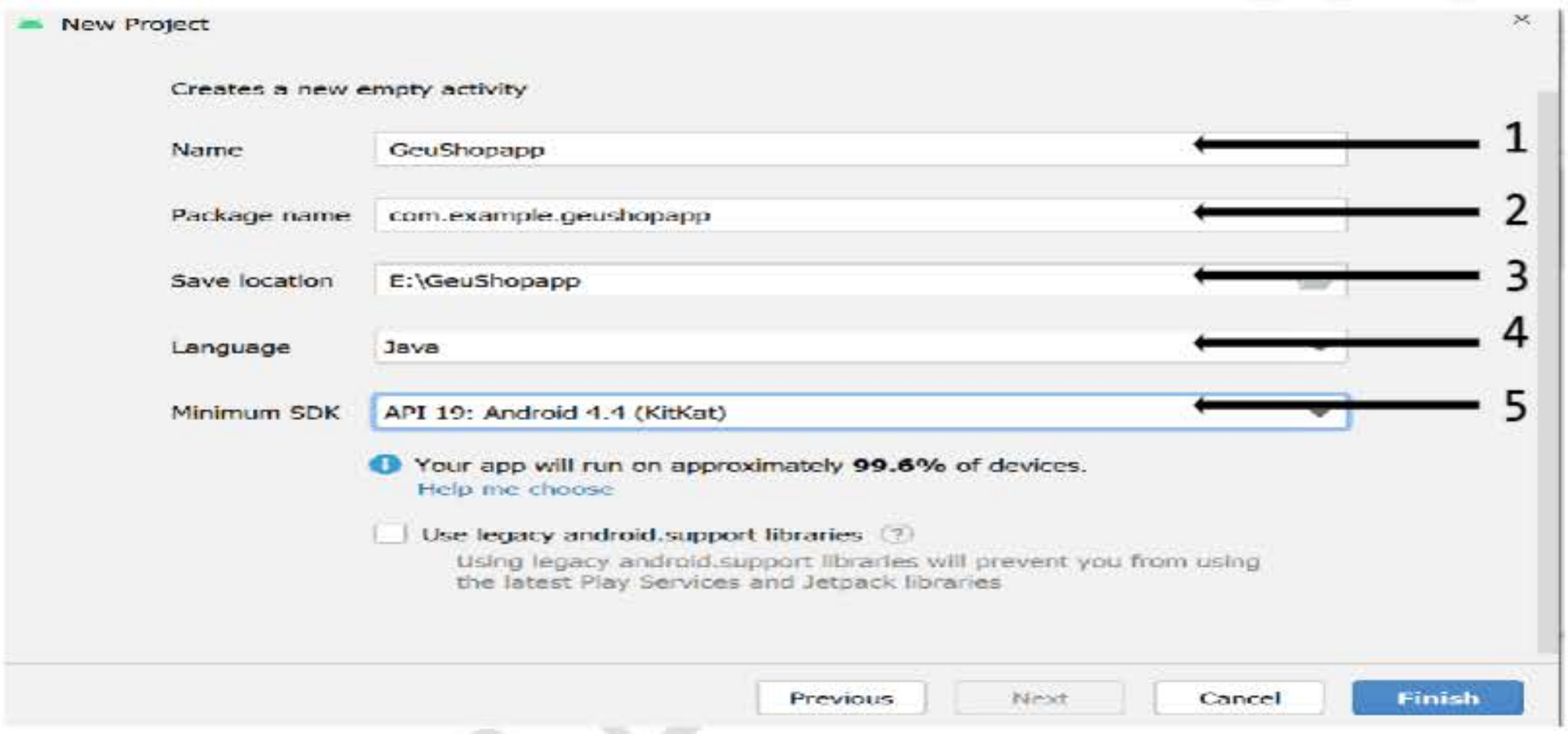

Figure 2.9: Empty activity window with Android Application name and other details

4. After some internal processing the files MainActivity.java, activity_main.xml, and AndroidManifest.xml are created with the default code.
    - MainActivty.java contains the java code. This is the entry point for an android application
    - activity_main.xml contains the default layout and user interface for an activity. It just contains one TextView control with the "Welcome to Graphic Era University" text set as the default value.
    - AndroidMainfest.xml contains information on all the components required for an android application

Android Studio Editor appears when the MainActivity.java file is opened. It contains the following code as shown in figure 2.10

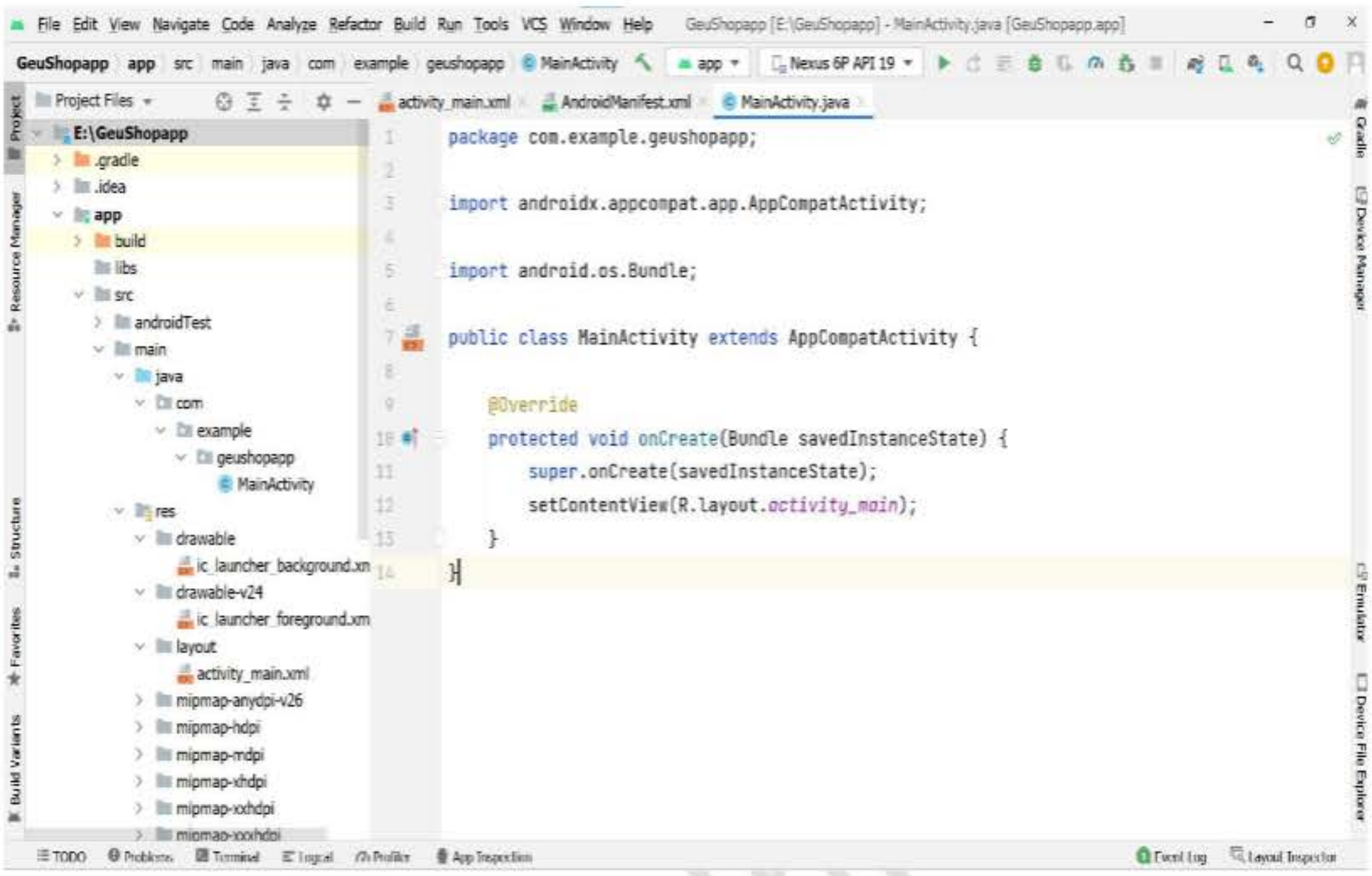

Figure 2.10: Default contents of MainActivty.java

when the activity_main.xml file is opened, it contains the following code as shown in figure 2.11

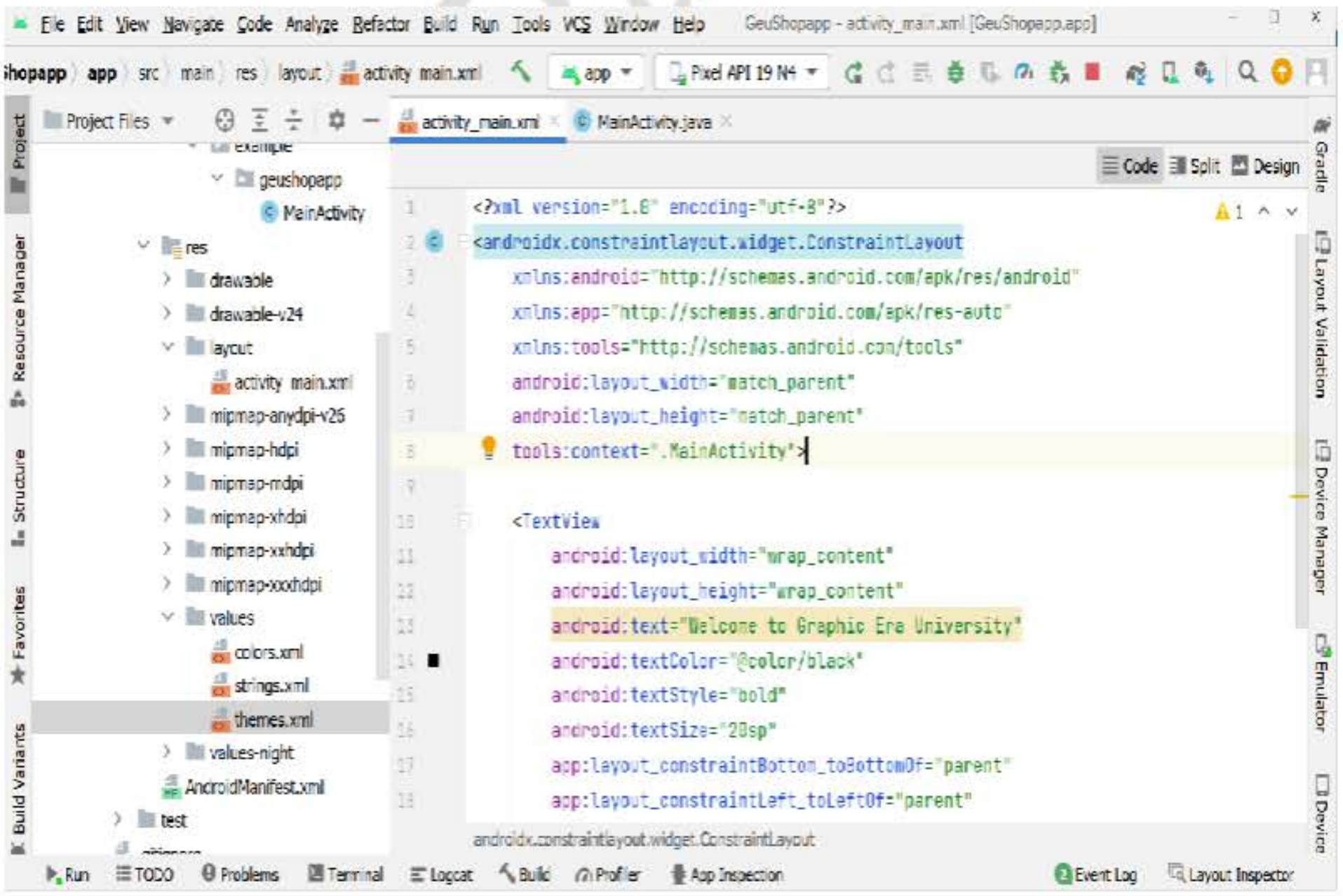

Figure 2.11: Contents of activity_main.xml

when the AndroidManifest.xml file is opened, it contains the following code as shown in figure 2.12

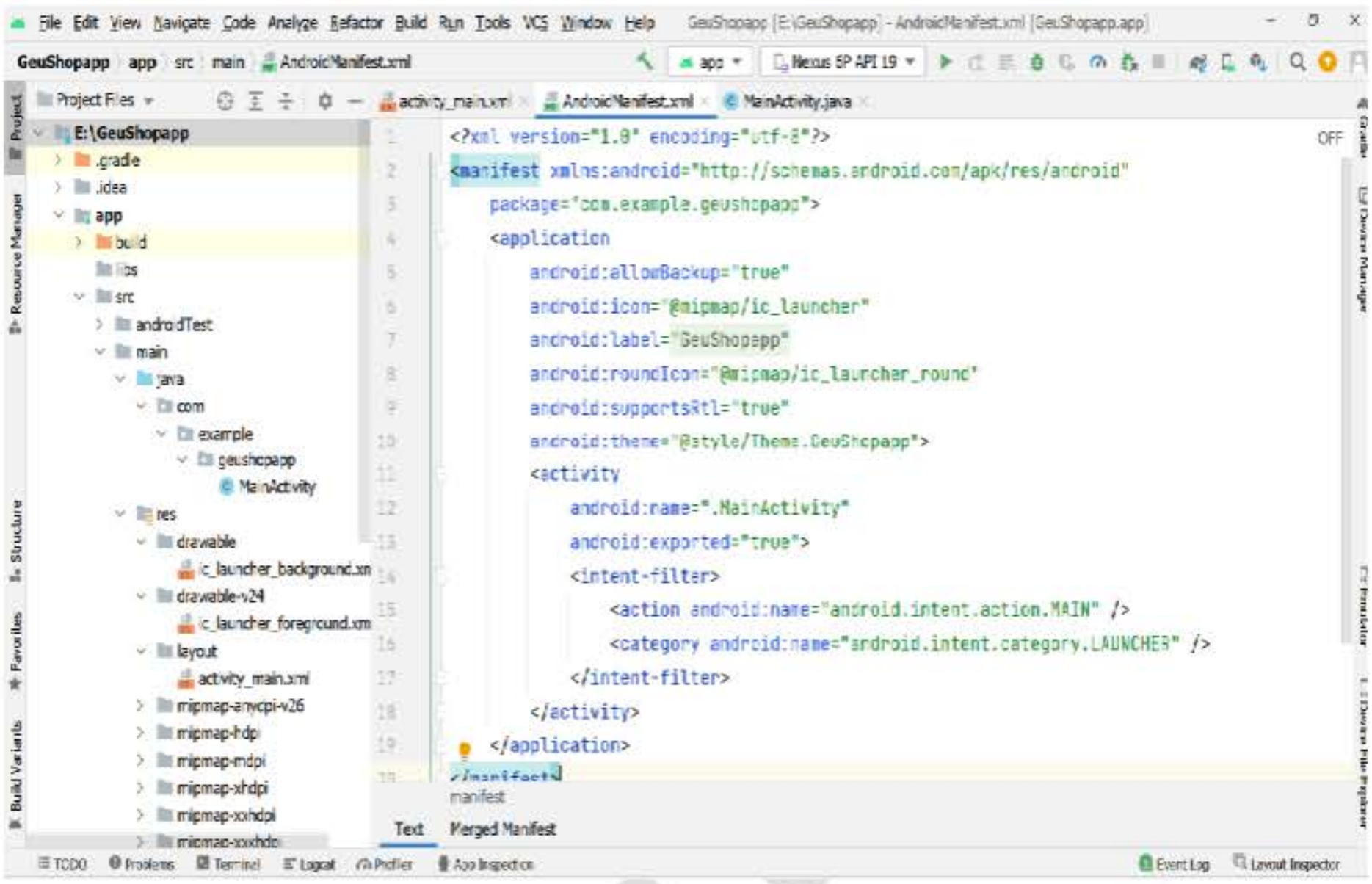

Figure 2.12: Contents of AndroidManifest.xml

5. To Run the application on the Android Virtual Device (Emulator)

In Android Studio select the virtual device from the drop-down (such as Pixel API 19 N4)

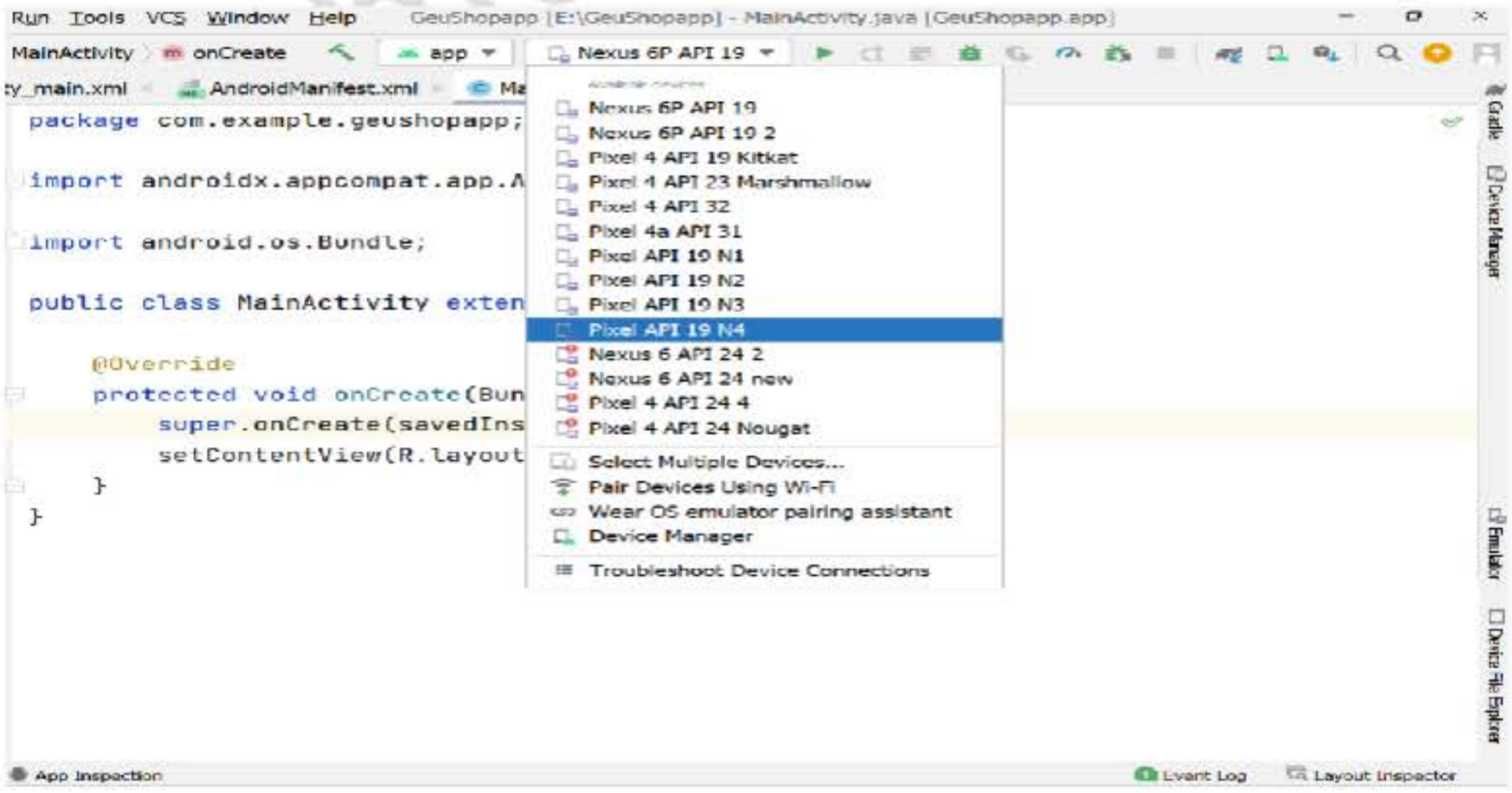

Figure 2.13: Select Pixel API 19 N4 (AVD) to run the application

6. In Android Studio, choose Run -> Run app or click the Run icon in the toolbar

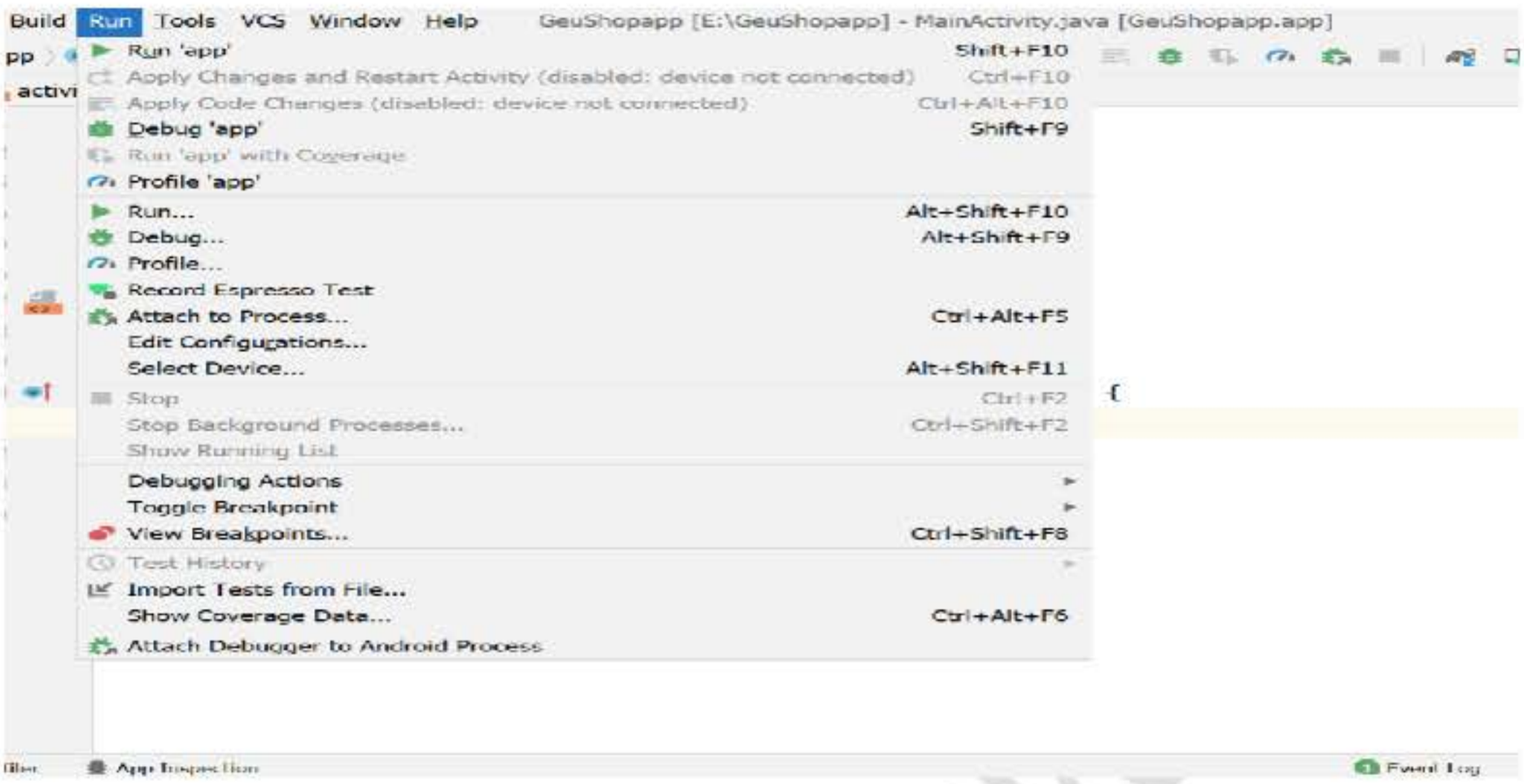

Figure 2.14: Select "Run app" to run the application

The emulator gets started and boots like a physical device. Once the emulator is ready the application is uploaded and executed in the emulator and the output appears as shown in figure 2.15

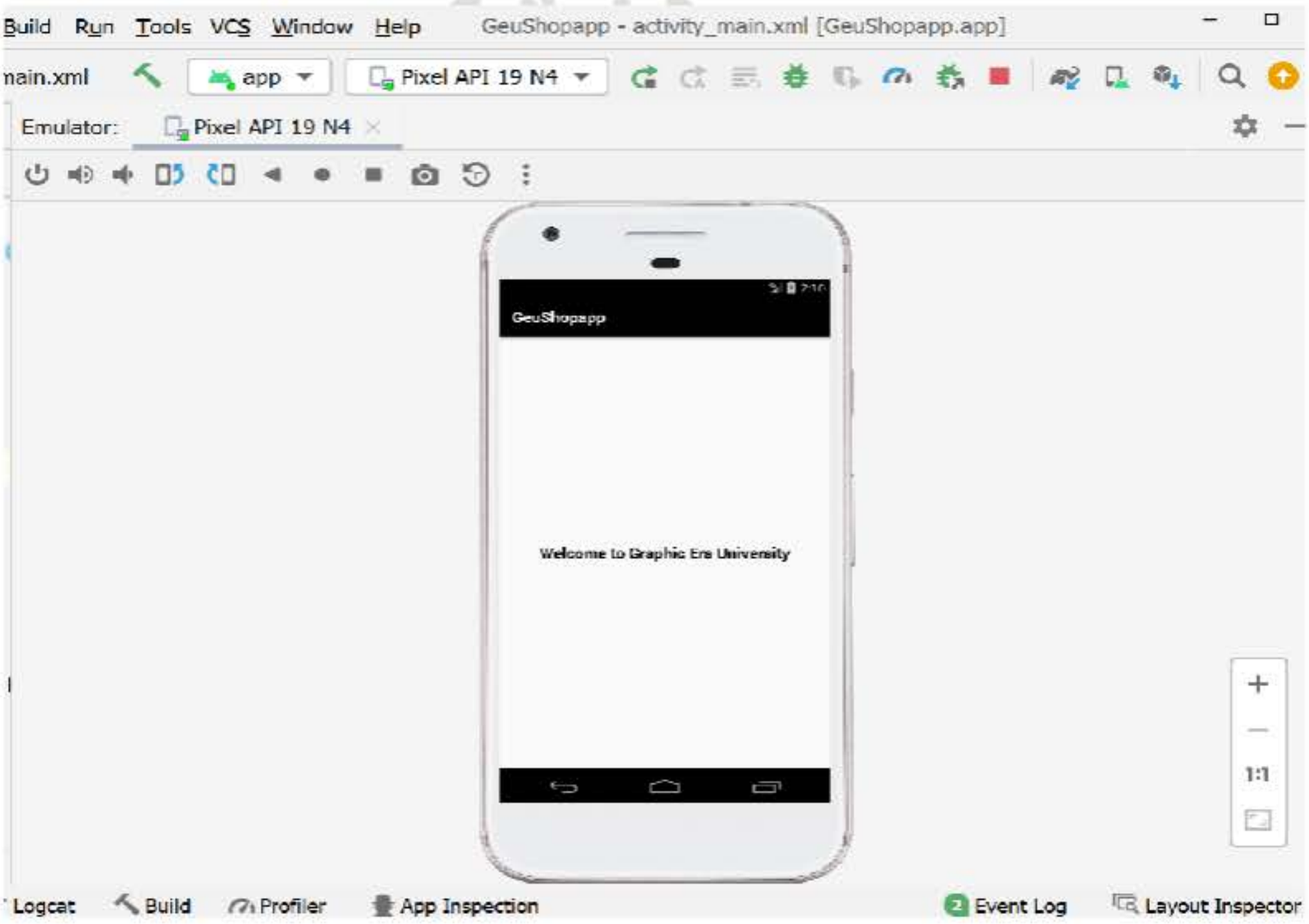

Figure 2.15: Output of the Android Application displayed on the Emulator

2.5 Self-Assessment Questions

Q1. Discuss the prerequisites, hardware, and software components required for installing Android Studio.[6 marks, L2]

Q2. What is an Android Virtual Device (AVD)? [2 marks, L2]

Q3. Explain the steps for creating an Android Virtual Device.[6 marks, L2]

Q4. Explain the steps of creating an android application to display a “Hello World” message.[8 marks, L2]

Q5. Make a list of files that are created by default in an Android project and explain their purpose of usage.[6 marks, L2]

2.6 Self-Assessment Activities

A1. Download and install JDK and Android Studio on your laptop or PC.

A2. Create Android Virtual Device (AVD) with different configurations which can be used for executing an Android application.

A3. Create an Android application to display the message “Welcome to GEU Shopping” and execute the application using different emulators.

2.7 Multiple-Choice Questions

1. The software to be installed before installing Android Studio is _____.[1 mark, L1]
   a) JDK
   b) Python
   c) C++
   d) None of the above

2. Android Studio can be installed on the _____ Operating System. [1 mark, L1]
   a) Windows
   b) Linux
   c) Mac OS X
   d) All of the above

3. The minimum RAM required to install Android Studio on Windows is __. [1 mark, L1]
   a) 2 GB
   b) 4 GB
   c) 8 GB
   d) 16 GB

4. The device used for testing an Android Application is ____. [1 mark, L1]
   a) Java Development Kit
   b) Android Virtual device
   c) Software Development Kit
   d) None of the above

5. The programming languages used in Android studio are _____. [1 mark, L1]
   a) JAVA and Kotlin
   b) JAVA and CSharp
   c) JAVA and Python
   d) All of the above

6. The default layout and the user interface is contained in ____. [1 mark, L1]
   a) activity_main.xml
   b) MainActivity.java
   c) AndroidManifest.xml
   d) None of the above

7. The java code is contained in _____. [1 mark, L1]
   a) activity_main.xml
   b) MainActivity.java
   c) AndroidManifest.xml
   d) None of the above

8. The file that contains information about all the components required for an android application is _____. [1 mark, L1]
   a) activity_main.xml
   b) MainActivity.java
   c) AndroidManifest.xml
   d) None of the above

9. An android application can run on ____. [1 mark, L1]
   a) Emulator
   b) Real device
   c) Only a)
   d) Both a) and b)

10. The file which is used as an entry point for executing an android application is ___.
    [1 mark, L1]
    a) activity_main.xml
    b) MainActivity.java
    c) AndroidManifest.xml
    d) None of the above

2.8 Key Answers to Multiple-Choice Questions

1. The software to be installed before installing Android Studio is JDK. [a]
2. Android Studio can be installed on Windows, Linux, and Mac OS X Operating Systems. [d]
3. The minimum RAM required to install Android Studio on Windows is 8GB. [c]
4. The device used for testing an Android Application is an Android Virtual device.[b]
5. The programming languages used in Android studio are JAVA and Kotlin. [a]
6. The default layout and the user interface are contained in activity_main.xml. [a]
7. The java code is contained in MainActivity.java.[b]
8. The file that contains information about all the components required for an android application is AndroidManifest.xml. [c]
9. An android application can run on an Emulator and a Real device. [d]
10. The file which is used as an entry point for executing an android application is MainActivity.java [b]

2.9 Summary

Depending on the operating system i.e. Windows, Linux, or mac OS before installing Android Studio the appropriate version of Java Development Kit (JDK) has to be installed as Java is the language used for developing an Android Application.

A minimum of 8GB RAM is required for running an android application in android studio using an android emulator, otherwise, the performance is very slow.

Select the appropriate platform based on the operating system used on the PC or laptop.

An Android Virtual Device (AVD) is used for testing an Android Application. We can create any number of AVDs with different configurations to test android applications.

Each AVD has different configurations which consist of hardware profile, emulated storage, system image, API level, and other properties.

An Android Emulator is an AVD that represents a specific Android device. An Emulator runs the Android operating system in an AVD and provides all the capabilities of a real Android device. We can simulate text messages, calls, device location, etc on an emulator. Running

android applications on emulators is faster than on a real device. Transfer of data is faster on an emulator.

When creating an Android application the files that are created by default are:

MainActivity.java, activity_main.xml, and AndroidManifest.xml

- MainActivty.java contains the java code. This is the entry point for an android application
- activity_main.xml contains the default layout and user interface for an activity.
- AndroidMainfest.xml contains information on all the components required for an android application

2.10 Keywords

- Java Development Kit (JDK)
- Android Virtual Device (AVD)
- Emulator
- MainActivity.java
- activity_main.xml
- AndroidManifest.xml

2.11 Recommended resources for further reading

Text Book(s)

1. Pradeep Kothari, "Android Application Development" Black-Book, Dreamtech Press,2015
2. Wei-Meng Lee, "Beginning Android Application Development", Wiley 2011.
3. Dawn Griffiths and David Griffiths, "Head First Android Development", O'Reilly, 2017.

References

- https://developer.android.com/guide
- https://developer.android.com/studio
- https://developer.android.com/ studio/install
- https://www.oracle.com/java/technologies/downloads/#jdk17-windows

--*--

Contents

UNIT 1 – Getting started with Android

CHAPTER 3 - Directory Structure and files of Android Application

---

Structure: Directory Structure and files of Android Application

---

3.1 Learning Outcomes

After the successful completion of this chapter, the student will be able to:

- Describe the directory structure of an Android Application
- Understand the basic files and the elements required for an Android Application

---

3.2 Directory Structure of Android Application

When an android application is created in the android studio many folders and files are generated.

- Among them, the resource folder is important as it contains non-code sources.
- The user interface for the android application is contained in the activity_main.xml file present in the layout sub-folder.
- Images and icons used by the application are present in a drawable sub-folder.

The directory structure and some important folders/files are marked/numbered as shown in figure 3.1

1. MainActivity.java file is contained in the java sub-folder of the source

   `\app\src\main\java\com\example\geushopapp\MainActivity.java`

2. activity_main.xml file is present in the layout sub-folder of the resource

   `\app\src\main\res\layout\activity_main.xml`

3. colors.xml is present in the values sub-folder of the resource

   `\app\src\main\res\values\colors.xml`

4. strings.xml is present in the values sub-folder of the resource

   `\app\src\main\res\values\strings.xml`

5. Androidmanifest.xml file is contained in the main sub-folder

   `\app\src\main\Androidmanifest.xml`

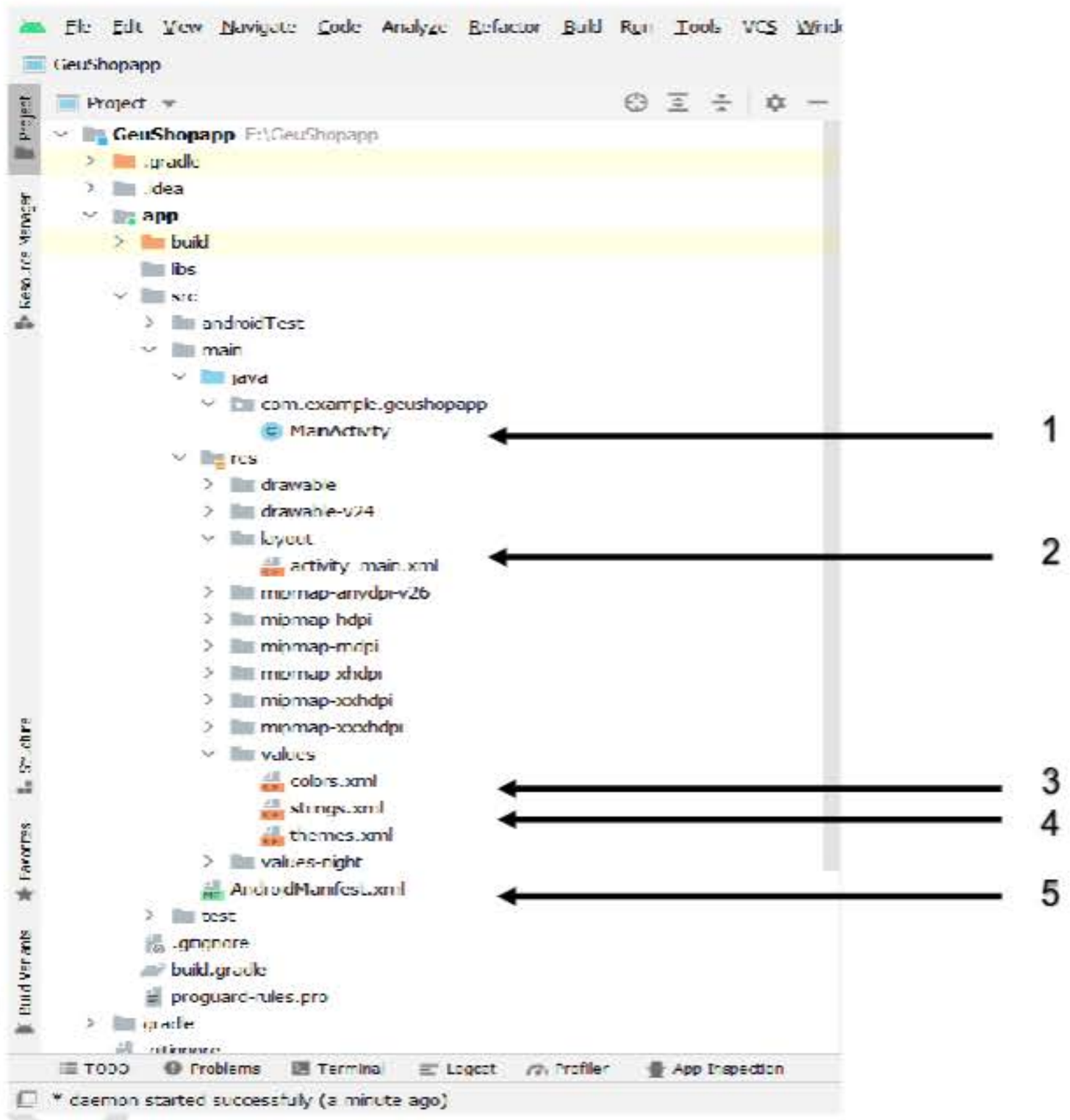

Figure 3.1: Directory structure of Android Application

3.3 File of Android Application

Many files are created by default when an android project is created. The following main files of interest are discussed here.

3.3.1 Android Manifest File: AndroidManifest.xml

- This file contains information/definitions of all the components used in an android application.
- Generally, an android application contains multiple activities.
- The definition of each activity used in the application, internet permission, SMS/MMS sending and receiving permission, database read/write permissions on external storage, minimum/maximum API level required to run an android application, package name, application launcher icon, main activity (entry point of an android application), etc are all stored in this AndroidMainfest.xml file.
- This acts as an interface between the Android Operating System and the application. Fundamental characteristics are described and defined in this file.
- <manifest> element being the root element, it consists of various other sub-elements like <application>, <activity>, <intent-filter>, <action> and attributes which specify the name of the activity, icon, and label of an application, etc.

The default code of AndroidManifest.xml generated by an android application on start is given below:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.geushopapp">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GeuShopapp">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figure 3.2: Snippet of AndroidManifest.xml file

Elements of AndroidManifest.xml file

Table 3.1: Elements of AndroidManifest.xml file

Elements	Description
<manifest>	It is the main/root element of the AndroidManifest.xml file in which all other elements are enclosed
<application>	It contains declarations of application activities and attributes that represent the icon, label, theme, etc.
<activity>	It contains the attributes that represent the name and label of the activity. It is a sub-element of <application>
<intent-filter>	<action> , <category>, <data> elements, etc are sub elements of <intent-filter>
<action>	At least one <action> element has to be included in <intent-filter>. The list cannot be empty
<category>	<category> elements are not mandatory to be included. It depends on the application
<data>	<data> element is used to specify MIME and URI

3.3.2 Main Activity File: MainActivtiy.java

The Main Activity is the entry point for the execution of an android application. It contains the code to handle all the activities in an android application. This is a java source file that is compiled by the java compiler into a class file and is later converted by the Dex compiler and executed by the Dalvik Virtual Machine.

The default code of MainActivty.java generated by an android application on start is given below:

```
package com.example.geushopapp;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }}
```

Figure 3.3: Snippet of MainActivity.java file

- The MainActivty class inherits from AppCompatActivity class.
- R.layout.activity_main – refers to the layout file activity_main.xml which contains the actual user interface.
- The onCreate() method is executed when the activity is first loaded.

3.3.3 Layout File: activity_main.xml

This file contains the layout of the user interface. This User Interface can be changed according to the requirements of the user.

By default, the activity_main.xml contains only TextView android control which is used to display a **"Hello World"** message when the application is executed. TextView control has various attributes like android:layout_width, android:layout_height, android:text, etc. The values can be set appropriately and a user-friendly interface can be designed. The layout can be changed to relative, or linear as per the user's requirements. Images to be used on buttons, height, width, text size, background color, background image, gravity, alignment to the top, bottom, left, right, below, above, etc of each control can be specified accordingly. The user interface of every activity used in the application is specified in the layout files. Each activity has a corresponding layout file associated with it.

The default code of activity_main.xml generated by an android application on start is given below:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
```

```
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Figure 3.4: Snippet of activity_main.xml file

3.3.4 Strings file: strings.xml

This file contains all the strings defined for use in an android application. The strings defined are given a unique name.

The default code of strings.xml generated by an android application on start is given below:

```
<resources>
    <string name="app_name">GeuShopapp</string>
</resources>
```

Figure 3.5: Snippet of strings.xml file

3.3.5 Colors file: colors.xml

Colors to use during different states in an application are defined in the colors.xml file.

Colors are specified either by using a hexadecimal or RGB value.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
</resources>
```

Figure 3.6: Snippet of colors.xml file

3.4 Self-Assessment Questions

Q1. Explain the directory structure of an Android Application.[8 marks, L2]

Q2. Explain the following files w.r.t android with an example. [5 marks each, L2]

- AndroidManifest.xml
- main.xml / activity_main.xml
- MainActivity.java

- strings.xml
- colors.xml

Q3. Explain the elements of AndroidManifest.xml. [7 marks, L2]

Q4. Explain the different files created by default in an android application and their purpose of use. [8 marks, L2]

3.5 Self-Assessment Activities

A1. Explore the following files and discuss the contents that can be added to them while developing an android application

- AndroidManifest.xml
- main.xml / activity_main.xml
- MainActivity.java
- strings.xml
- colors.xml

3.6 Multiple-Choice Questions

1. AndroidManifest.xml contains _____. [1 mark, L1]
   a) All information about an application
   b) Only information about the layout in an application
   c) Only information about activities in an application
   d) None of the above

2. Layout files are stored in _____ directory. [1 mark, L1]
   a) /res/values
   b) /src
   c) /res/layout
   d) /assets

3. The root element in the AndroidManifest.xml file is_____. [1 mark, L1]
   a) <application>
   b) <manifest>
   c) <root>
   d) <information>

4. Images and icons used in an android application are stored in a ___ folder.[1 mark, L1]
   a) drawable
   b) layout
   c) values
   d) None of the above

5. <intent-filter> should have at least one ____ element. [1 mark, L1]
   a) <data>
   b) <category>
   c) <action>
   d) <activity>

6. The file that contains user interface elements is _____. [1 mark, L1]
   a) AndroidManifest.xml
   b) MainActivity.java
   c) activity_main.xml
   d) strings.xml

7. The file that contains all the strings defined for use in an android application is _____.
   [1 mark, L1]
   a) AndroidManifest.xml
   b) MainActivity.java
   c) activity_main.xml
   d) strings.xml

8. The method which is executed when the activity is first loaded is _____. [1 mark, L1]
   a) onCreate()
   b) onLoad()
   c) main()
   d) None of the above

9. An android application can contain _____ activities. [1 mark, L1]
   a) only one or two
   b) multiple
   c) only two
   d) maximum 5

10. Colors to use during different states in an application are defined in _____. [1 mark, L1]

a) strings.xml

b) colors.xml

c) MainActivity.java

d) AndroidManifest.xml

3.7 Key Answers to Multiple-Choice Questions

1. AndroidManifest.xml contains all information about an application.[a]
2. Layout files are stored in /res/layout directory.[c]
3. The root element in the AndroidManifest.xml file is <manifest>.[b]
4. Images and icons used in an android application are stored in a drawable folder.[a]
5. <intent-filter> should have at least one <action> element.[c]
6. The file that contains user interface elements is activity_main.xml.[c]
7. The file that contains all the strings defined for use in an android application is strings.xml.[d]
8. The method which is executed when the activity is first loaded is onCreate().[a]
9. An android application can contain multiple activities.[b]
10. Colors to use during different states in an application are defined in colors.xml.[b]

3.8 Summary

When an android application is created in the android studio many folders and files are generated.

- Among them, the resource folder is important as it contains non-code sources.
- The user interface for the android application is contained in the activity_main.xml file present in the layout sub-folder.
- Images and icons used by the application are present in a drawable sub-folder.

AndroidManifest.xml - This file contains information/definitions of all the components used in an android application. Generally, an android application contains multiple activities.

MainActivtiy.java - The Main Activity is the entry point for the execution of an android application. It contains the code to handle all the activities in an android application. This is a java source file that is compiled by the java compiler into a class file and is later converted by the Dex compiler and executed by the Dalvik Virtual Machine.

activity_main.xml - This file contains the layout of the user interface. This User Interface can be changed according to the requirements of the user.

strings.xml - This file contains all the strings defined for use in an android application. The strings defined are given a unique name.

colors.xml - Colors to use during different states in an application are defined in the colors.xml file. Colors are specified either by using a hexadecimal or RGB value.

3.9 Keywords

- AndroidManifest.xml
- activity_main.xml
- strings.xml
- colors.xml
- drawable
- resource folder

3.10 Recommended resources for further reading

Text Book(s)

1. Pradeep Kothari, "Android Application Development" Black-Book, Dreamtech Press,2015
2. Wei-Meng Lee, "Beginning Android Application Development", Wiley 2011.
3. Dawn Griffiths and David Griffiths, "Head First Android Development", O'Reilly, 2017.

--**--