

CHAPTER 1

INTRODUCTION

Mobile application development is the process to making software for smartphones and digital assistants, most commonly for Android and iOS. The programming and markup languages used for this kind of software development include Java, Swift, C#, and HTML5. Talking about the mobile applications, the first thing that comes to mind are the apps like Whatsapp, Instagram, swiggy, etc that we use in our everyday life. Mobile app development is rapidly growing.

1.1 Different ways to develop Mobile apps:

There are 3 different ways to develop Mobile apps:

1. 1st Party Native App development
2. Progressive web Application
3. Cross-Platform Application

1.2.1 1st Party Native App development:-

These types of apps normally run in the native devices, that is, it runs only in the OS that it is specifically designed for it. These apps cannot be used on different devices using a different OS. The apps that are developed for android are normally coded using Java or Kotlin languages. The IDE normally used for android app development is Android Studio which provides all features and the apps that are developed for IOS are generally coded in swift language or Objective-C. The IDE suggested for IOS App Development is XCode.

Advantages:

- The performance of these apps are very high these apps very fast compared to any other apps.
- We have easy access to all the features and APIs.

Disadvantages:

- The development speed is too slow as we have to code it again for different OS.
- And this category doesn't support open source.

1.2.2 Progressive web Application:

Progressive web apps are essentially a website which runs locally on your device. The technologies used are Microsoft Blazor, React, Angular JS, Native Script, Ionic. These technologies normally used for web development propose. The apps' UI is developed the same way as they are developed while developing the website. This category has many ups and downs let's start with the advantages of Progressive web apps.

Advantages:

- The main advantage of this process is that its development speed is fast the same code base is used for IOS, Android, web applications.
- The web development team can be repurposed to develop the mobile application.
- No installation required.

Disadvantages:

- The major disadvantage is that PWA don't have access to all the feature and so the user experience is not that good IOS does not support all the features of PWA
- The UI for development is bespoke i.e.the buttons, edit texts need to be programmed which was not necessary for the 1st party native Apps.
- The community is not the wide spread.
- No extra room for business model i.e.it is still challenge to develop a revenue model or advertising opportunities for PWAs. At the moment, there are fewer options than among native apps to subscribe to.

virtually indispensable. Scientific visualization became an important field in the 1980s when the scientists and engineers realized that they could not interpret the prodigious quantities of data produced in supercomputer runs without summarizing the data and highlighting trends and phenomena in various kinds of graphical representations.

1.2.3 Cross-Platform Application:

These are frameworks that allow developing total native applications which have access to all the native features of IOS and Android but with the same code base. These apps run on both Android and IOS. So normally the development speeds of these apps are very fast and the maintenance cost is low. The performance speed is comparatively low to 1st party native apps but faster than PWA.

Advantages:

- The apps' development speed is very high as they use the same code base for both Android and IOS.
- The apps' maintenance cost is low as the errors and updates as to be countered only once.

Disadvantages:

- Slow Code Performance With Limited Tool Availability.
- Limited User Experience i.e. these apps does not have access to Native only features.

CHAPTER 2

SYSTEM REQUIREMENTS SPECIFICATION

System requirements are intended to communicate in precise way, the functions that the system must provide. To reduce ambiguity, they may be written in a structured form of natural language supplemented by tables and system models.

2.1 HARDWARE REQUIREMENTS

The physical components required are:-

- ✓ Processor: Intel core i3 processor
- ✓ RAM: 8GB
- ✓ Secondary Memory: 500GB

2.2 SOFTWARE REQUIREMENTS

The software used in building this app are as specified:-

- ✓ Operating system-Android Studio
- ✓ Language-Java,XML
- ✓ JDK Version-Java Development Kit(JDK) 7 or higher device independent package. the program should be capable to run on any screen resolution.

2.3 FUNCTIONAL REQUIREMENTS

CHAPTER 3

INTRODUCTION TO ANDROID STUDIO

3.1 DESCRIPTION

The Android operating system is most frequently used on different mobile platforms around the world. It is occupied approximately 75% of shares in the worldwide market by the end of 2020. A company like Open Handset Alliance has developed the first Android that depends on the customized version of the Linux kernel as well as other open-source software. At the initial stage of 2005, Google sponsored the project & it obtained the entire company. In September 2008, the first Android device was released in the market to dominate the mobile industries due to several features like user friendly, the support of the community is huge, customization, manufacturing of android devices in large companies. Consequently, the market examines the demand to develop Android-supported devices with smart developers. So, the Android operating system became a complete set of operating systems for different devices like wearables, mobiles, notebooks, smart TVs, tablets, set-top boxes, etc.

3.2 What is an Android Operating System?

Android is a Linux-based operating system it is designed primarily for touch screens mobile devices such as smartphones and tablet computers. The operating system has developed a lot in the last 15 years starting from black and white phones to recent smartphones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.



Android Operating System

Android is a powerful operating system and it supports a large number of applications on Smartphones. These applications are more comfortable and advanced for users. The hardware that supports android software is based on the ARM architecture platform. The android is an open-source operating system that means that it's free and anyone can use it. Android has got millions of apps available that can help you manage your life one or another way and it is available at low cost in the market for that reason android is very popular.

Android development supports the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of the android development kit (SDK) was released in 2008 and the latest updated version is a jelly bean.

3.3 Features of Android Operating System

The unique features/**characteristics of the android operating system** include the following.

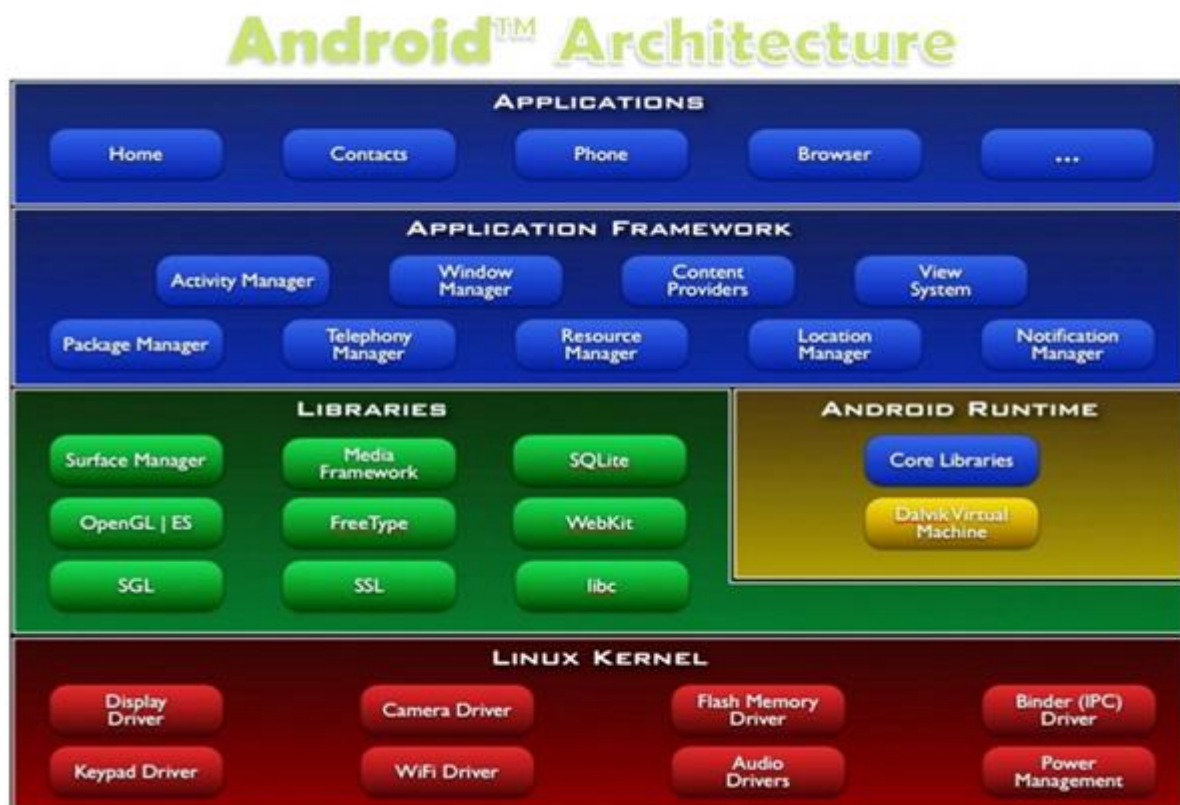
- Near Field Communication (NFC)
- Alternate Keyboards
- IR Transmission
- No-Touch Control
- Automation
- Wireless App Downloads
- Storage & Battery Swap
- Custom Home Screen
- Widgets
- Custom ROMs
- Headset layout
- Storage
- Connectivity: GSM/EDGE, IDEN, CDMA, Bluetooth, WI-FI, EDGE,3G,NFC, LTE,GPS.
- Messaging: SMS, MMS, C2DM (could to device messaging), GCM (Google could messaging)
- Multilanguage support
- Multi-touch
- Video calling

- Screen capture
- External storage
- Streaming media support
- Optimized graphics

3.4 Android Architecture

The android is an operating system and is a stack of software components which is divided into five sections and four main layers that is

- Linux kernel
- Libraries
- Android runtime
- Application Framework
- Applications



Android Architecture

3.4.1 Linux Kernel

The android uses the powerful Linux kernel and it supports a wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from the software. This provides basic system functionalities like process management, memory management, device management like camera, keypad, display, etc the kernel handles all the things.

Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hardware devices on the system.

3.4.2 Libraries

The on top of a Linux kernel there is a set of libraries including open-source web browsers such as WebKit, library libc. These libraries are used to play and record audio and video. The SQLite is a database that is useful for the storage and sharing of application data. The SSL libraries are responsible for internet security etc.

3.4.3 Android Runtime

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in java language. The Dalvik VM enables every Android application to run its own process. The Dalvik VM executes the files in the .dex format.

3.4.4 Application Framework

The application framework layer provides many higher-level services to applications such as windows manager, view system, package manager, resource manager, etc. The application developers are allowed to make use of these services in their applications.

3.4.5 Applications

You will find all the android applications at the top layer and you will write your application and install it on this layer. Examples of such applications are contacts, books, browsers, services, etc. Each application performs a different role in the overall applications.

3.4.6 Android Emulator

The Emulator is a new application in the Android operating system. The emulator is a new prototype that is used to develop and test android applications without using any physical device.



Android Emulator

The android emulator has all of the hardware and software features like mobile devices except phone calls. It provides a variety of navigation and control keys. It also provides a screen to display your application. The emulators utilize the android virtual device configurations. Once your application is running on it, it can use services of the android platform to help other applications, access the network, play audio, video, store, and retrieve the data.

3.4.7 Android Versions

The different **android versions** from the beginning to the present are mentioned below from 1.0 to 2.0. Here is the **android operating system list**.



Android Versions from 1.0 – 1.1

The first Android version like 1.0 was released in the year 2008 but it didn't have any codename.

Android Version – 1.5

In the year 2009, the 1.5 android version was released which is named Cupcake.

Android Version – 1.6

The Android version – 1.6 was released in the year 2009 which is called Donut. The ability of this version is, it operates on different screen sizes as well as resolutions.

Android Versions from 2.0 – 2.1

Éclair 2.0 version was released after six weeks of the Donut version.

Android Version – 2.2

After four months of 2.1 version invention, the Android 2.2 version was released which is called Froyo.

Android Version 2.3

The first true visual identity of Android version 2.3 was released in the year 2010 namely Gingerbread.

Android Version from 3.0 – 3.2

In the year 2011, the versions from 3.0 – 3.2 were released which is named Honeycomb

Android Version 4.0

Android 4.0 was released in the year 2011 which is called Ice Cream Sandwich.

Android Versions 4.1 – 4.3

Android Versions 4.1 to 4.3 were released between 2012 to 2013 which is called Jelly Bean.

Android Version 4.4

Android Version 4.4 was released in Late-2013's which is called KitKat.

Android 5.0

The version of Android 5.0 was launched in the year 2014, Nov in the Nexus 6 device which is called Lollipop

Android 5.1

This version of Android 5.1 was released in March 2015 called Lollipop

Android 6.0

Android 6.0 Version like Marshmallow was released in Oct 2015.

Android 7.0

Android 7.0 version was released by Google in Aug 2016 called Nougat.

Android 7.1, 7.1.1 & 7.1.2 Nougat

Android 7.1 version was released in Oct 2016, called Nougat

Android 8.0

Android 8.0 version was released in Aug 2017 called Oreo.

Android 8.1.0

Android 8.1.0 version was released in Dec 2017, called Oreo

Android Version 9 Pie

Android version 9 Pie was released in Aug 2018

Android Version 10

Android version 10 was launched in September 2019

Android Version 11

Android version 11 was released recently on September 8th, 2020

3.5 ADVANTAGES

- Android is a Linux based open-source operating system, it can be developed by anyone
- Easy access to android apps
- You can replace the battery and mass storage, disk drive, and UDB option
- Its supports all Google services
- The operating system is able to inform you of a new SMS and Emails or the latest updates.
- It supports Multitasking
- Android phone can also function as a router to share the internet
- It's free to customize
- Can install a modified ROM
- Its supports 2D and 3D graphics
- We can install Millions of apps-
- Backup and restore of apps can be possible
- It supports Third-party apps
- Addition & removal of unwanted features:-
- High job demands for Android developers
- Notifications can be displayed very clearly
- Huge community support
- The Internet can be shared from device to device

- It is an Open source
- Different types of mobile models you can select
- Add/ Remove Unwanted
- Internal memory is Expandable
- Cloud storage
- It supports big screens at a reasonable price
- Foldable Android devices
- Different apps can run at the same time
- Several widgets on display

3.6 DISADVANTAGES

- Apps work in the background
- Battery discharges easily due to a lot of processes within the background.
- Requires Google account
- Less security, so fake apps can be easily installed to steal your data from strange resources
- Mobiles with low specification run very slow
- Generally, you require additional code on Java language as compared to Objective-C.
- Android developers have a critical time
- Difficult layouts & animations are tough to code within Android.
- Protection of Virus is required
- For developers of the app, Google is strict
- Several ads within apps
- Some apps quality is not good
- For elders, it is not friendly

CHAPTER 4

INTRODUCTION TO JAVA

Android-based smartphones are in vogue due to the flexibility they offer for customization. Unlike Apple's iOS, Google Android offers better user experience in terms of applications. The Android application development kit is an open-source Linux-based operation system, which has its own middleware and key applications. The platform for app development in Android is Java. This means that you use the Java library and code the applications in Java, C, and C++ programming language. But, the most widely used programming language for android application development is Java.

4.1 Why Java for Android application development

If you want to get started with application development, Google provides a Java API to get started and compiles your files into classes. Why did Android prefer Java for its development platform? There are multiple reasons such as; Java is a commonly used language and many programmers know it, it can run on a virtual machine (VM) so no need to recompile for different phones, better security, many development tools available for Java, and Java is a known industry language with most phones compatible with it.

Though Google provides the Java API, Android does not use JVM to execute class files. Rather, it uses Dalvik Virtual Machine (DVM). The class files are compiled into Dalvik Executable (DEX) format, and bundled as Android Package (APK) along with other resources.

With Java, if you are aware of object-oriented programming principles, creating applications for android will be much simpler than iOS app development.

4.2 PRE-REQUISITES

Before you can begin Java programming for Android, you will need certain tools installed. Ensure that you download the Android SDK Bundle, which includes the Android SDK and Integrated Development Environment (IDE). After you download the bundle, unzip the

contents and double-click the **SDK Manager.exe** file. Once the installation is complete, from the **Start** menu, start **Eclipse IDE** that was bundled with the SDK.

4.3 DEVELOPING AN APP USING JAVA FOR ANDROID

For expert level users of Java, programming for Android will not be difficult. Let us understand how one can go about developing an application using a step-by-step approach.

1. Creating an Android project in the Eclipse IDE. Here you specify certain project options such as name of the project, what Android version the project will run on, app name, class name, package name and so on.
2. Configure the project and select a launcher icon. This means you are providing workspace details and selecting the appropriate size of the launcher icon.
3. Creating activities is a very important aspect of app development. This activity is nothing but different activities a user gets to do on-screen. After you are done with selecting the type of activities, the project is now open with relevant resource files for you to begin the actual Java for android coding.

The user experience elements and the look of your app is defined using the **activity_main.xml** file in the **/res/layout** folder from the package. In this file, you will modify the string attributes and add views for the app. You can also add radio buttons or text fields based on the app requirement and design. By right-clicking on the element you added, you can select properties and modify them based on your layout.

4.4 IMPLEMENTING LOGIC USING JAVA

After the front-end elements are finalized, the most important step remains implementing logic for all the activities to work well. The logic needs to be implemented in the **MainActivity.java** file from the **src/com.example.tutorialapplication/** folder. The **MainActivity.java** file is the file that actually get converted into Dalvik compatible format and runs the application.

Let us consider that you created the front end to have a text field that displays “Udemy Online Courses” and here is the code snippet for this text display in the application.

This code snippet is for the **MainActivity.java** file.

```
package com.example.udemy;

import android.os.Bundle;

import android.app.Activity;

import android.view.Menu;

import android.view.MenuItem;

import android.support.v4.app.NavUtils;

public class MainActivity extends Activity {

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

    @Override

    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.activity_main, menu);

        return true;

    }

}
```


In the above snippet, the **R.layout.activity_main** element calls the activity file.

The **onCreate** and **onCreateOptionsMenu** are one of the many methods that are executed when **MainActivity.java** file executes.

4.5 DECLARING YOUR COMPONENT IN ANDROID MANIFEST FILE

The **AndroidManifest.xml** file found in the root of the application or project folder is where you need to declare your package for your application to execute successfully. This file bridges the gap between the OS and your application. Here is the code snippet to declare your package in the manifest file.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.udemy"

    android:versionCode="1"

    android:versionName="1.0" >

    <uses-sdk

        android:minSdkVersion="8"

        android:targetSdkVersion="15" />

    <application

        android:icon="@drawable/ic_launcher"

        android:label="@string/app_name"

        android:theme="@style/AppTheme" >

        <activity

            android:name=".MainActivity"
```

```
android:label="@string/title_activity_main" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER"/>

</intent-filter>

</activity>

</application>

</manifest>
```

In the above snippet, all the application related resources are provided under the **<application>...</application>** tags. This includes the application icon, theme, app name, the name of the activity, and so on. The basic declaration made in the manifest file are **<activity>** for activities in the app, **<service>** for all the services used by your class, **<receiver>** for broadcast receiver, and **<provider>** for content providers.

4.6 PROVIDING TEXT ELEMENT IN THE STRINGS FILE

All the textual elements used in the UI of the app are provided in the Strings file. This includes button names, labels, default text on-screen, and so on. The following code snippet shows the default string declarations:

```
<resources>

<string name="app_name">Udemy Online Courses</string>

<string name="udemy_courses"> Udemy Online Courses!</string>

<string name="menu_settings">Settings</string>
```

```
<string name="title_activity_main">MainActivity</string>

</resources>
```

4.7 FINALIZING THE LAYOUT OF YOUR APPLICATION

The **activity_main.xml** mentioned earlier is used to control the layout of the app. This means you will be altering it quite often to make your app screen optimized. The following code snippet shows default layout declarations:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout_width="match_parent"

android:layout_height="match_parent" >

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:layout_centerHorizontal="true"

android:layout_centerVertical="true"

android:padding="@dimen/padding_medium"

android:text="@string/udemy_courses"

tools:context=".MainActivity" />

</RelativeLayout>
```

4.8 RUNNING THE CREATED APPLICATION

For executing your application, you must install the Android Virtual Machine (AVM). This lets you execute your application on your computer, virtually. Click the **Run** icon in your Eclipse IDE to execute your application.

CHAPTER 5

RESULT

The following screens show the design of our mini project

CHAPTER 7

CONCLUSION

REFERENCES

- [1] Erik Hellman, "Android Programming-Pushing the Limits",1st edition, Wiley India Pvt Ltd,2014.
- [2] Dawn Griffiths and David Griffiths, "Head First Android Development",1st edition, O'Reilly SPD Publishers,2015.
- [3] J F DiMarzio, "Beginning Android Programming with Android Studio",4th edition, Wiley India Pvt Ltd,2016.
- [4] Anubhav Pradhan, Anil V Deshpande, "Composing Mobile Apps" using Android, Wiley 2014.