Abstract

The aim of this project was to investigate the use of Android and Anti Theft Alarm together by developing an Android application which made use of mobile users. The specific function of the application was to allow the authorised user of the mobile to access if not, any intruder or bad guy is access without a permission this makes the alert sound to make attraction of the mobile user and then quickly fetch the real time information for that intruder or bad guy who enters the wrong pattern or password to access the phone.

The technologies used in this project are the Android Software Development Kit (SDK) and Kotlin. The Android SDK is used to take care of general Android application development and Kotlin is a programming language is used to create an app easy and fast.

CHAPTER 1

INTRODUCTION

This chapter will outline the aims and motivation behind the project and a general overview of both the project itself and this report.

1.1 AIMS

The aim of this project was to investigate Android studio, which is a relatively new technology. This is an Android port of computer vision libraries to do image manipulation.

In order to do this investigation, I developed an Android application which uses techniques for an anti theft alarm libraries to secure the phone from the criminals or thief to avoid stealing for that phone users.

1.2 MOTIVATION

A lot of modern research goes into maximizing connection speeds in developed countries so that

They have very little use as other than when someone new to that system (people who recently moved residence, tourists, etc.) needs them to tell whether they have kept the phone carefully or not even they are in crowded places.

In cases where there exists a Real Time anti theft mobile system, information could be capture from the user who given, which could then be used to obtain the real time information for the mobile user. Nowadays, more and more people carry smartphones with them wherever they go, so developing an application which would do this is quite feasible.

Anti theft alarm has many techniques which allows this to be possible. Even more so now because it is now possible to use anti theft techniques in both Android and iOS applications.

1.3 PROJECT OVERVIEW

For this particular project I looked at the real time information for Android mobiles. Android mobile have real time information for all of its android OS.

However, crowded area of the city there exists no such awareness of keeping our mobile safety, and if mobile user would like to know how to keep our mobile safety from bad guys in the crowded area, they would have to open up the Anti Theft Alarm app, go to the relevant search option, and input the type of security they want to keep the mobile safety. This can take a while as, depending on what the mobile user does, the Anti Theft Alarm app could end up making multiple calls to the Anti Theft API.

I created an Android application which would allow the passenger to simply take a picture of the intruder who access your phone without your permission, and the application would save the intruder photo from it and fetch the real time information. Once it reads the password given by the intruder and also it matches with the password saved by the mobile user from this once the matching false, it makes a alert tone to the Anti Theft API which returns the loud sound to the mobile user.

1.4 OVERVIEW OF REPORT

The next chapter will describe the System analysis for the project

Then, Chapter 3 will talk about system specification of the application.

Chapter 4 will briefly describe background of android and the components I the market level.

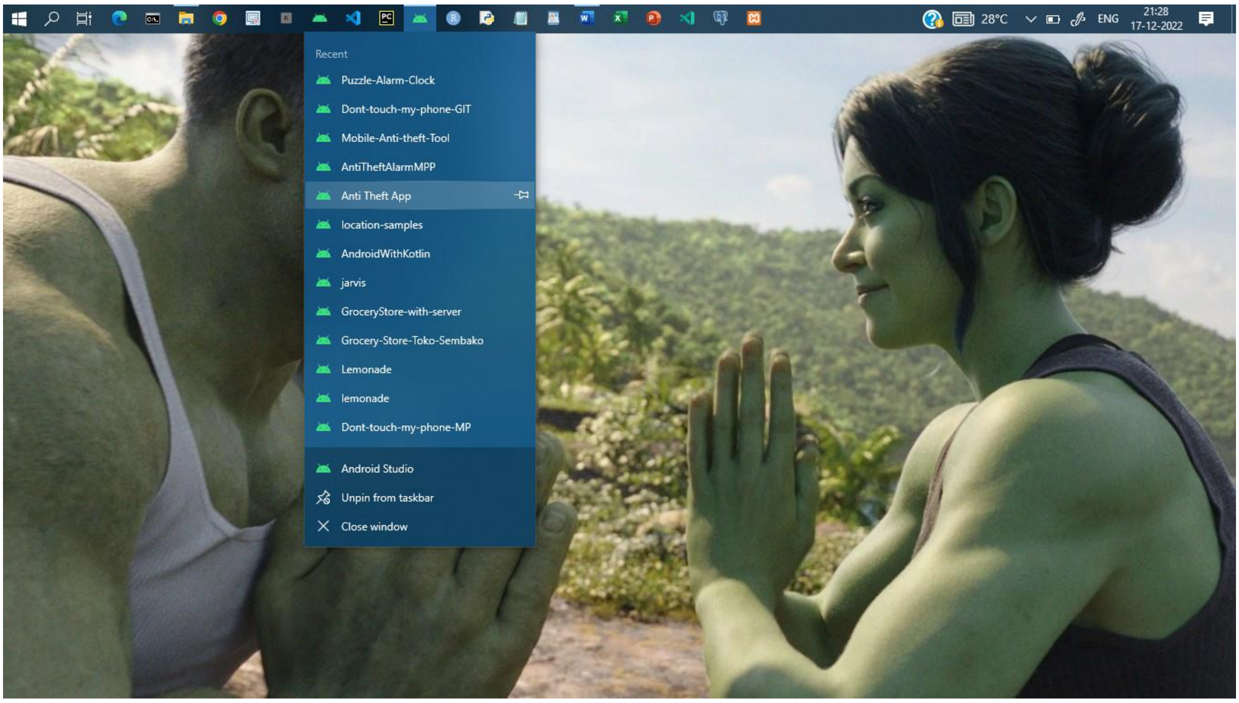
Chapter 5 will briefly describe about design choices for each step of the application and how each of these steps work with each other.

Chapter 6 will briefly describe some possibilities for future development of the application, including possible features that it is currently lacking.

Finally, there will be a ‘Conclusion’ chapter which will summarise the results of my project and what I was able to achieve from doing it.

CHAPTER 2

SYSTEM ANALYSIS



2.1 EXISTING PROJECT

MOTION ALARM – App detects motion when someone tries to touch it without your permission

DISCONNECT CHARGER ALARM – Protect your phone while it’s charging an alarm will sound when someone tries to disconnect the charger cable

PIN CODE – For super protection

FINGERPRINT – Switch off the alarm with your fingerprint

ALARM SOUNDS – Choose any alarm you want

RUNS ON EVERY ANDROID PHONE – Another strong point of this app is that it runs well even on older Android devices and is generally undemanding in terms of system resource usage.

2.2 PROPOSED SYSTEM

PROXIMITY ALARM – App detects when someone tries to take out your phone from the pocket without your permission

SECURITY CODE – Password, Pin code, Pattern, Fingerprint

FINGERPRINT – When the user places his/her finger in the sensor panel while pickup it turned off the alert. If not registered fingerprint is detected while pickup it turned on the alert sound

FLASHLIGHT DETECTION – When someone pickup our phone in the dark room at that time our phone is in silent mode it shows the alert by blinking the mobile flashlight

INTRUDER SELFIE – When someone tries to unlock our phone without our permission at that time the camera is running in the background and capture the intruder face. It also capture the intruder that how many times he /she fails to succeed in breaking the password and shows the timing of this action is done.

CHAPTER 3

SYSTEM SPECIFICATION

3.1 SOFTWARE SPECIFICATION

ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for android application development. Android studio provides more features that enhance our productivity while building Android apps. It has flexible Gradle-based build system. Fast and feature rich emulator for app testing. It makes easy to integrate Google Cloud Messaging and App Engine. We can create multiple APKs from our app, with different features using the same project and modules.

KOTLIN

Kotlin is an open-source, statically-typed programming language that support both object-oriented and functional programming. Kotlin provides similar syntax and concepts from other languages. It exits in variants that target the JVM, JAVASCRIPT, and native code. Kotlin is official supported by Google for Android Development, meaning that Android documentation and tooling is designed with Kotlin. Kotlin's interoperability with java is core to its growth. It means that we can call into java code from Kotlin and vice-versa.

JAVA

Being an auto-generated file that is generated by AAPT, Android R.java contains resource IDs for all the resources of res/ directory. The id for the created component is automatically generated in the R.java whenever a component is created in the android activity_main.xml file. The life cycle methods for an activity such as onCreate, onStop, onResume, etc is provided by the Activity java class. The created ID can later be used in the java source file. TO act on a component, the corresponding id can be used in the activity source file. The android R.java file contains many static nested classes such as menu, id, layout, attr, drawable, string, etc.

XML

The Android platform uses xml files in projects for many purposes, from providing basic configuration of the application in the Manifest file, to using XML Layout files to define the user interface. These standard files in the same way they are used when working with the java language, developers have access to the same controls and UI capabilities as all other Android developers, fully natively. Any changes we make to your XML layouts in Android Studio will automatically sync back into your project, and elements defined in your layouts and the other XML files will be available via the static R class in your project’s default namespace.

PLATFORM: ANDROID STUDIO

FRONT – END: KOTLIN, JAVA, C/C++

BACK – END: XML

3.2 HARDWARE SPECIFICATION

To implement these minimum configurations of the hardware support is required. Hardware specification means the minimum technical specification and configuration that must be met by the Hardware in order to ensure the correct operation of the Software, as set out in the Documentation.

Processor: 1GB RAM and 2GHZ Dual core processor and above.

Minimum 1280 * 800 resolution and above

CHAPTER 4

BACKGROUND

In this chapter about the technologies required to create the application for which this report is about, specifically about Android development itself, and libraries required for anti theft alarm.

4.1 ANDROID

In this project, Android refers to the mobile operating system which is developed by Google and is based on the Linux Kernel. Its main use is for touchscreen mobile devices such as smartphones and tablets, but it can also be used to make specialised user interfaces for televisions and wrist watches.

In July 2013, the Google Play store hit the one million mark for the number Android applications published, with over 50 billion downloaded of these apps. In 2014, Google announced that there were over one billion active monthly Android users, which is approximately double what it was the previous year.

Google have released Android’s source code under open source licenses, which means that it is free for everybody to access. This has resulted in a larger community of developers using the open-source code as a foundation for community-driven projects. Android’s success has gotten it involved with the “smartphone wars” between technology companies.

4.1.1 HISTORY

The Android operating system originally started its life with Android, Inc., which was a small company founded by Andy Rubin, Rich Miney, Nick Sears, and Chris White,

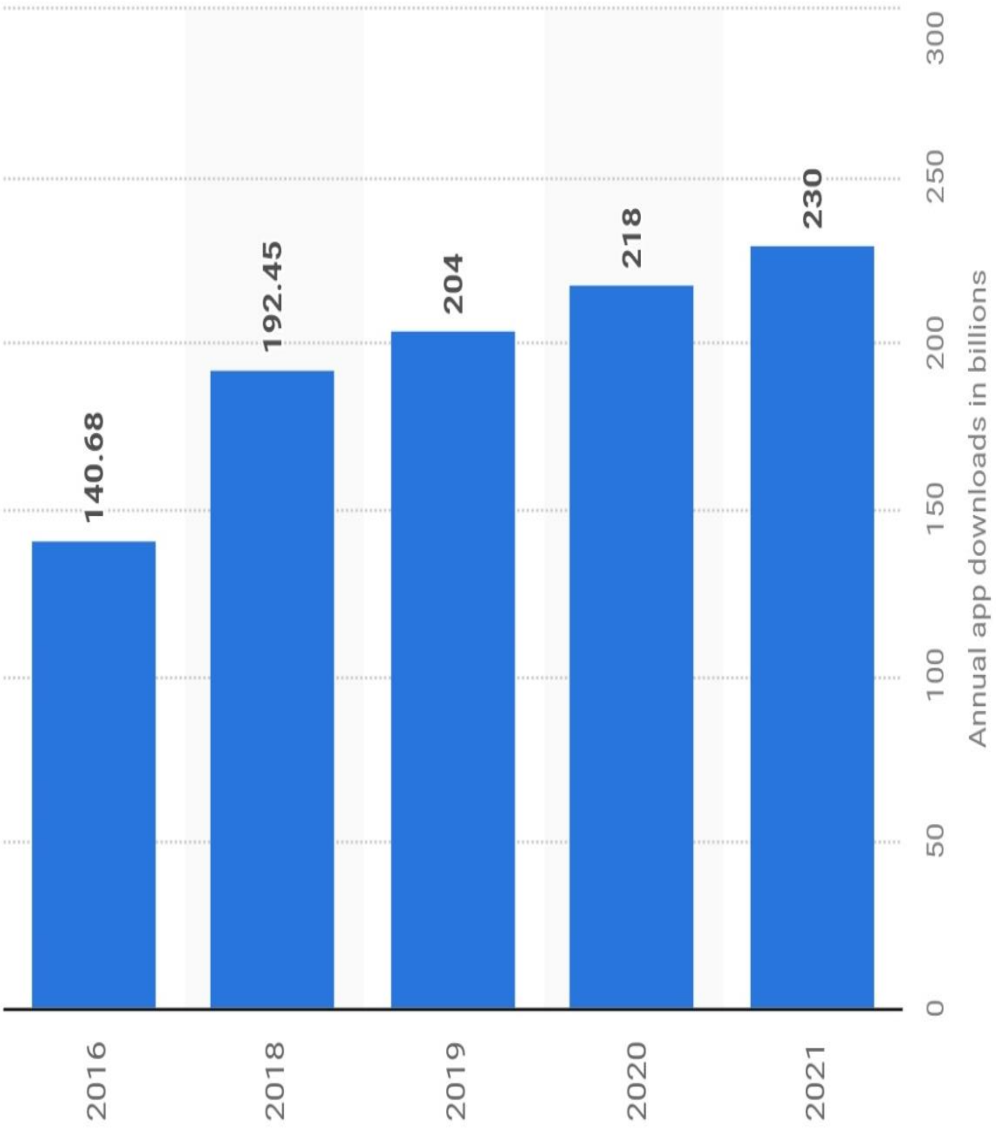


Figure 1.1: Number of app downloads from Google Play from 2016 to 2021

in October 2003. The company’s original aims where to make an advanced operating system for digital cameras to help improve the connection between cameras and PCs. Despite the fact that the founders of Android, Inc. had been very successful in the past, they decided to operate the company in secret, only saying that they were working on software for mobile phones.

In August 2005, Google bought Android, Inc. with a number of the original founders staying with the company. Rubin then led a team which developed a mobile device platform using the Linux Kernel. Later, towards the end of 2007, Android was finally unveiled by the Open Handset Alliance, with the first smartphone running Android being the HTC Dream which was released the following year.

There have been many updates to Android since its release in 2008, both for bug fixes and for the addition of new features. So far there have been 10 major versions of Android, with the latest one being Android 5.0 “Lollipop”. Figure 2.2 shows the distribution of Android versions being used at a given point between December 2009 and March 2015. Every major version of Android has been named in alphabetical order after a dessert or sugary treat.

4.1.2 DEVELOPMENT

The development of Android versions and updates are done in private by the team at Google. It is only when it is ready to be release to the public that the source code is made available to the public. Google aims to have a major update to Android every six to nine months.

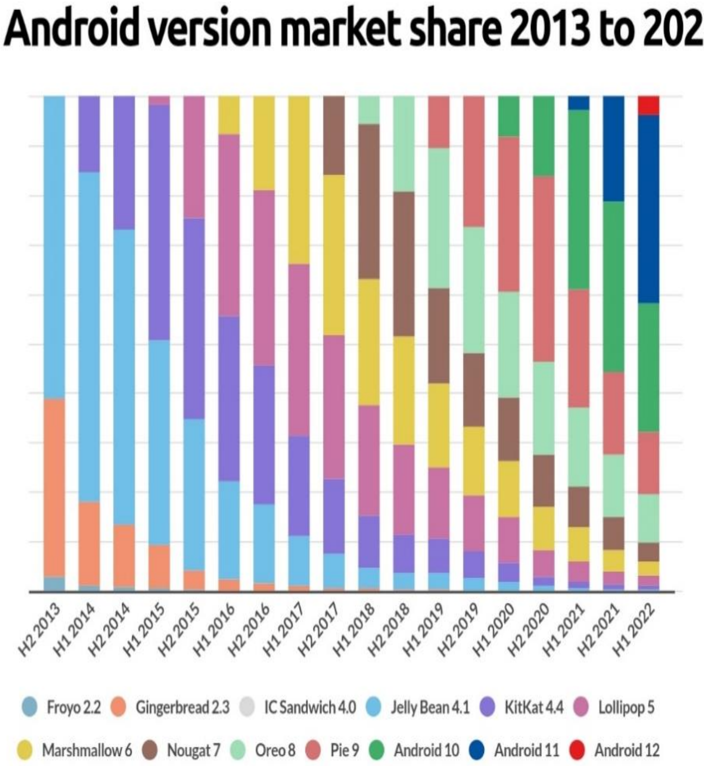


Figure 1.2: Distribution of Android versions throughout the world between 2013 and 2022

However, depending on the device in question updates can be slow to reach it. Being Google's own product and the flagship device for Android, the Nexus brand of phone and tablet receive updates as soon as they are released. For other brands, it can take a couple of months until the update is available. The reasoning for this is that there can be an extensive variation in hardware between different Android devices, so each update needs to be tailored for that device.

The kernel used by Android is based on the Linux kernel's long-term support branches, generally being either version 3.4 or 3.10. The kernel version varies depending on the actual Android device.

4.2 ANDROID STUDIO

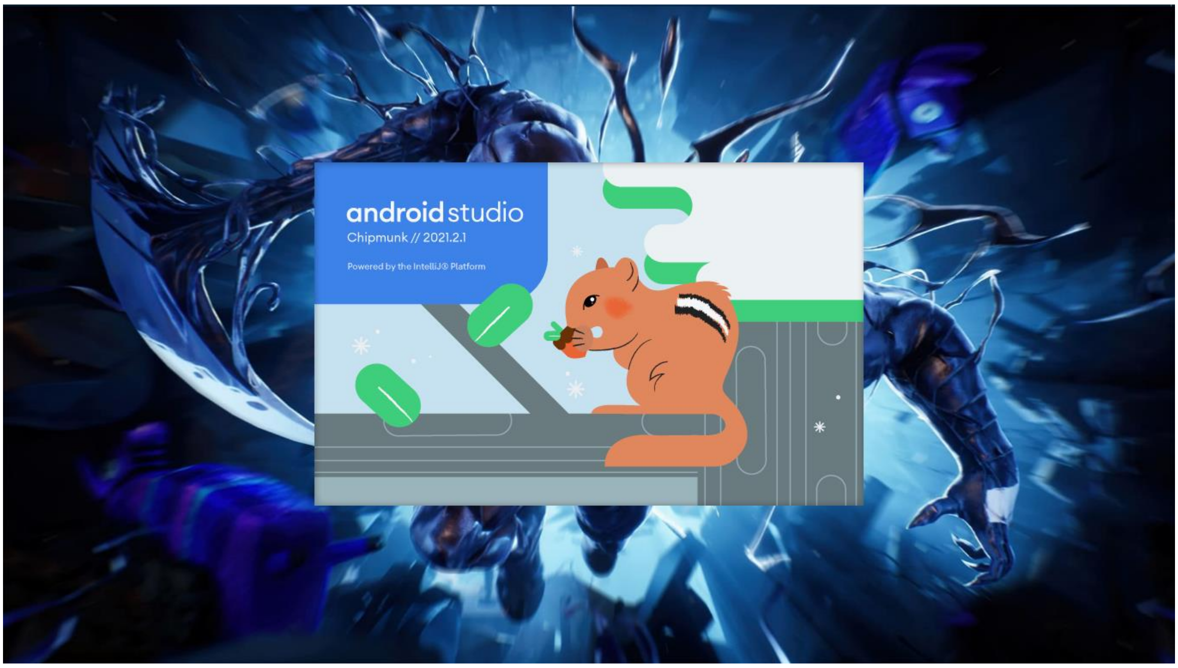
An Android Studio is the official integrated development environment for android application development. Android Studio was announced on 16th May 2013 at the Google I/O conference. It started its early access preview from version 0.1 in May 2013. The first stable built version was released in Dec 2014, starts from version 1.0. Since 7th May 2019, Kotlin is Google's preferred language for Android application development. The android Studio project one or more modules with resource files and source code siles.

These include different types of modules.

Android app modules

Library modules

Google App Engine modules



This view is formed by modules to provide quick access to our project's key source files.

MANIFESTS – It contains the AndroidManifest.xml file

JAVA – It contains the source code of java files, including the Junit test code.

RES – It contains all non-code resources, UI strings, XML Layouts, and bitmap images.

4.3 KOTLIN

Kotlin is an open-source, statically-typed programming language that supports both object-oriented and functional programming. Kotlin provides similar syntax and concepts from other languages.

DEVELOPERS – Everybody who used Kotlin once within a given period

ACTIVE DEVELOPERS – Developers who edited Kotlin code in both the current and previous periods. Active monthly users, people who edited code both this month and last month.

August 2020–July 2021: 4.86 million users

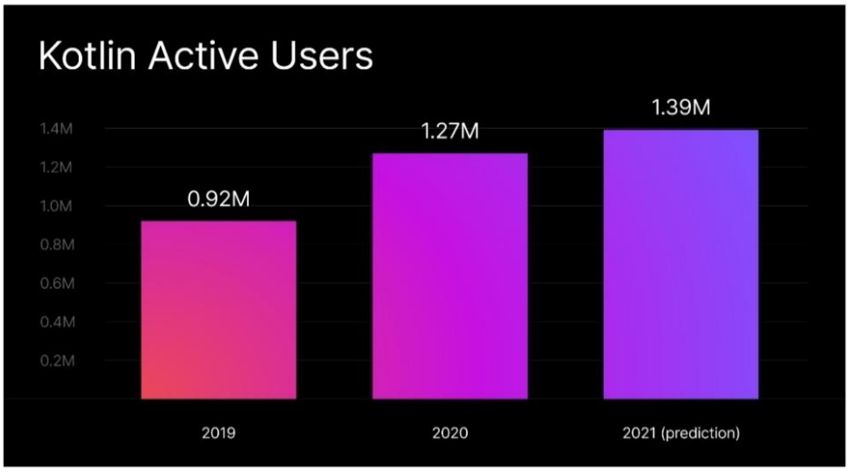


Figure 1.3: Distribution of Kotlin active users throughout the world between 2019 and 2021

August 2020 –July 2021: 1.32 million active users

According to this user of Kotlin is increased gradually, in 2019, range is increased to 0.92 million and the next year 2020, range is increased to 1.27 million and finally, the previous year 2021 is increased by 1.39 million.

In the last 12 months, we have had 4.86 million Kotlin developers globally. Of these users, 1.32 million were active developers in at least one month. The average number of active developers per month in 2021 is 373,000.

The general amount and the share, and that allows us to estimate professional audience size. For Kotlin, it is 1-1.9 million developers. An annual survey conducted by the JetBrains MR&A team. This report is based on the input of 31,743 developers from 183 countries and regions. Based on the survey results and the estimation of the professional developers population, about 2 million professionals have used Kotlin at least once in 2021. Among them, roughly 1 million professional developers use Kotlin as one of their 3 primary languages.

4.4 CONCLUSION

To summarize, there are a number of different technologies required to produce this application. Even though it hasn’t been around for too long, the Android operating system continues to grow more and more every year. With this increase, so too does the interest in the development of applications for the operating system. The same could be said for the use of android studio, even if it has been around for slightly longer. Despite the fact that Android development and recognition are two completely different fields, a community exists to try and bring the two together, as shown by the fact that the Android version of the Recognition libraries have been downloaded over 370,000 times in the last 3 years.

CHAPTER 5

Design and Implementation

This chapter about the design choices and how I implemented the application. First about the development of the Android application itself, then to an individual techniques used for different parts of it.

5.1 ANDROID DEVELOPMENT

The development of Android applications is done using the Android Software Development Kit (SDK). This SDK comes with all the necessary development tools needed, including all the relevant libraries, a debugger and an Android emulator. The main platforms which are currently supported for development or Android are any modern desktop Linux distribution, Windows XP or later, and Mac OS X 10.5.8 or later.

The languages used in Android development are Java, Kotlin and XML. The entire back-end functionality, including the use of the Recognition libraries, is done in Kotlin. XML is used for both the front-end design & layout of the pages, and for the general configuration of the application.

5.2 APP COMPONENTS

5.2.1 PROXIMITY DETECTION

It is a sensor that detects when the phone is take out from the packet. The alert sound is rings aloud. Our body has electromagnetic force when, the force is applied on the mobile the electromagnetic field is disturbed at that time the alert is rings.

5.2.2 MOTION DETECTION

It is a sensor that detects when the phone is picked up from the surface. Sensor detects motion by measuring changes in the amount of sound or radiation reflecting back into the receiver. When an object interrupts or alters the sensor's field, at that time the alert is rings

5.2.3 CHARGING DETECTION

It is a sensor when the phone is disabled from charging. The alert sound is rings aloud. When the charging port is disconnected by a bad guy or an intruder at that time mobile got alert.

5.2.4 KIDS ZONE

It is a app lock that the user can lock the apps that the user want to give security for a particular or specific apps he/she wants.

5.3 Android Activities

Android applications have components known as "activities". Activities are essentially the different pages that you would see. Each of these have both a Kotlin file which handles all of the back-end functionality, Java Virtual Machine (JVM) which helps the app to run, and a corresponding XML file which handles the front-end design & layout.

AN android activity goes through six major lifecycle stages or call backs.

onCreate()

onStart()

onResume()

onPause()

onStop()

onDestroy().,

this system invokes each of these call backs as an activity enters a new state.

LIFE CYCLE OF AN ANDROID

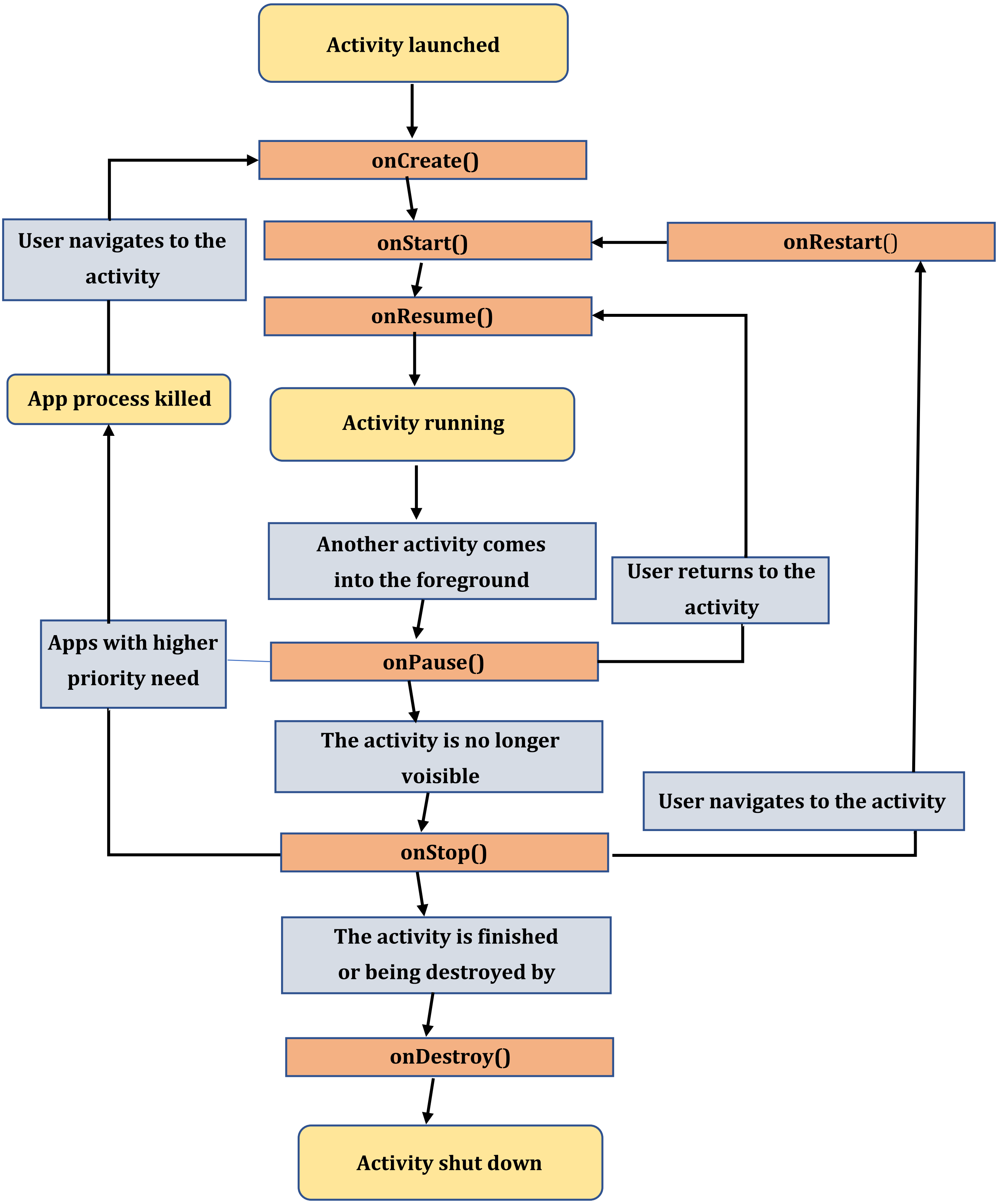
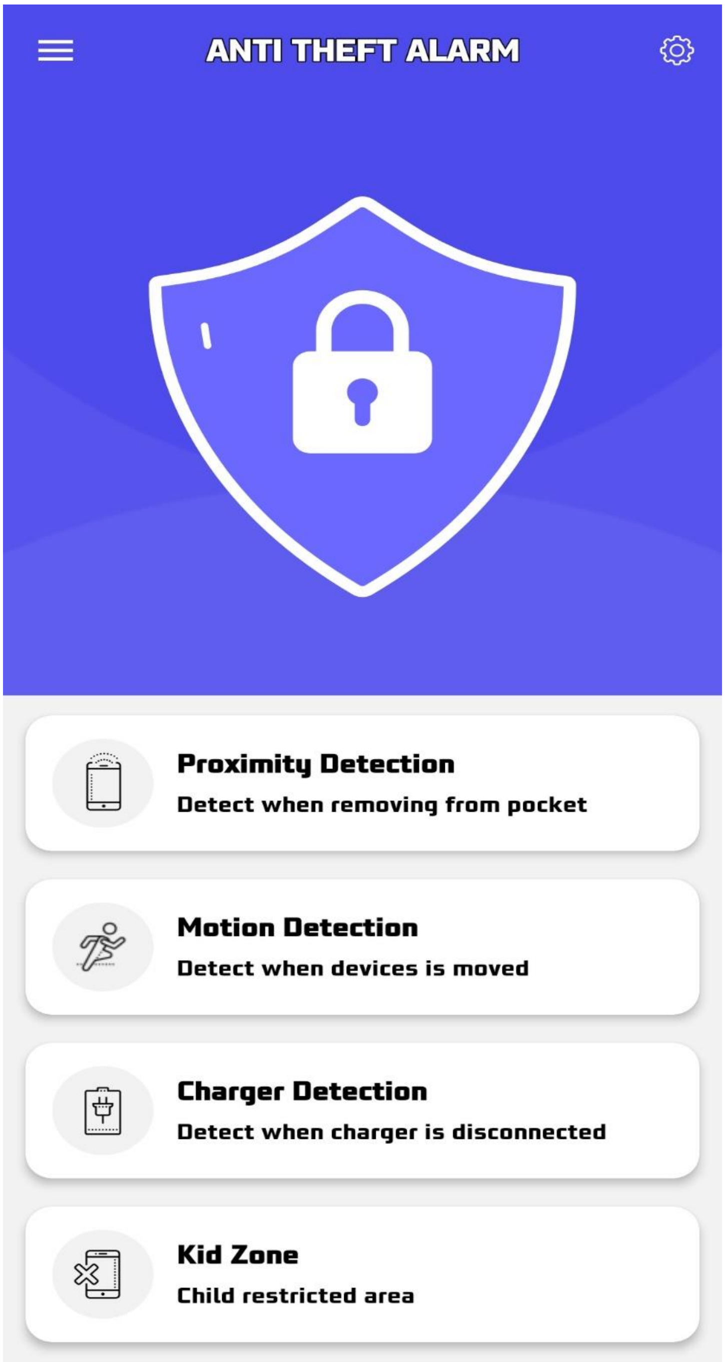

Activity launched
onCreate()
User navigates to the activity
onStart()
onRestart()
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need
onPause()
The activity is no longer voisible
User navigates to the activity
onStop()
The activity is finished or being destroyed by
onDestroy()
Activity shut down

Figure 1.4 LIFE CYCLE OF AN ANDROID APPLICATION DEVELOPMENT

5.3.1 MAIN ACTIVITY

This is a very basic activity which acts as a landing page. It is what gets displayed when the user first opens the app. When the user selects the Motion Detection button, the activity will launch an activity which loads the activation page and mobile user can able to activate. After this, it will return to the main activity with the result action. This will then make a alert to the user when some one passed into it.



ANTI THEFT ALARM

Proximity Detection
Detect when removing from pocket

Motion Detection
Detect when devices is moved

Charger Detection
Detect when charger is disconnected

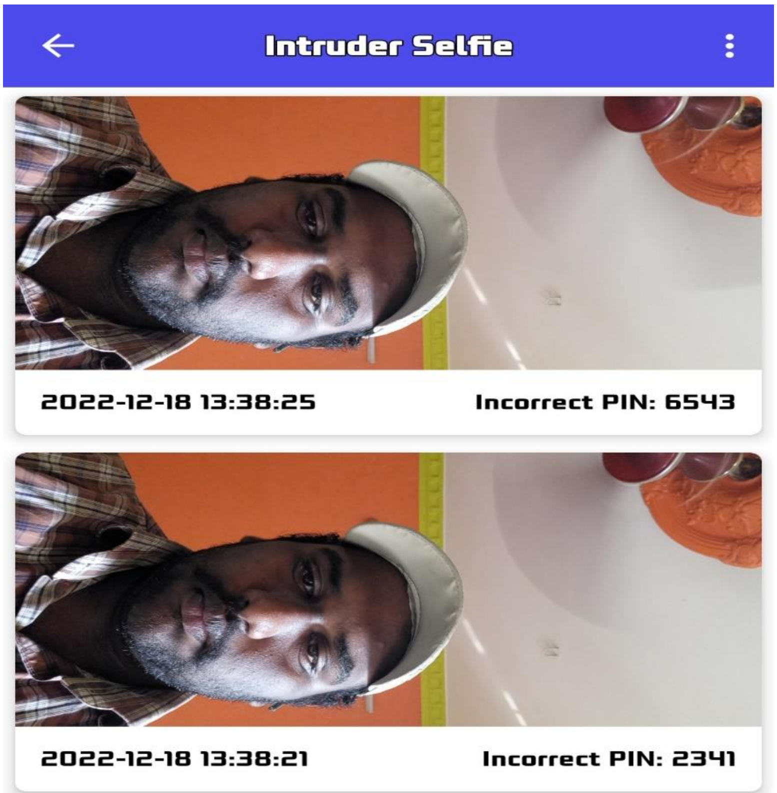
Kid Zone
Child restricted area

Figure 1.5: Screenshot of the Main activity

5.3.2 IMAGE ACTIVITY

As I mentioned previously, this activity gets launched after the bad guy tries to unlock the mobile at that time it takes a picture with a camera. All of the image processing takes place in this activity.

Before doing this, the activity sets the front camera image to be displayed on the background, and it saves to the file for the user to view. It will also show the user at what time and how many times the unknown guy is tries to unlock the mobile from the image, also letting the user confirm it.



2022-12-18 13:38:25 Incorrect PIN: 6543

2022-12-18 13:38:21 Incorrect PIN: 2341

Figure 1.6: Screenshot of the Image activity

The user can simply use the Back button which will take them back to the landing page, and they can re-set the image file.

5.3.3 RESULTS ACTIVITY

Finally, we have the results activity. It starts with the passcode activity area. The user can then return back to the previous activity, allowing them to go through the whole process again.

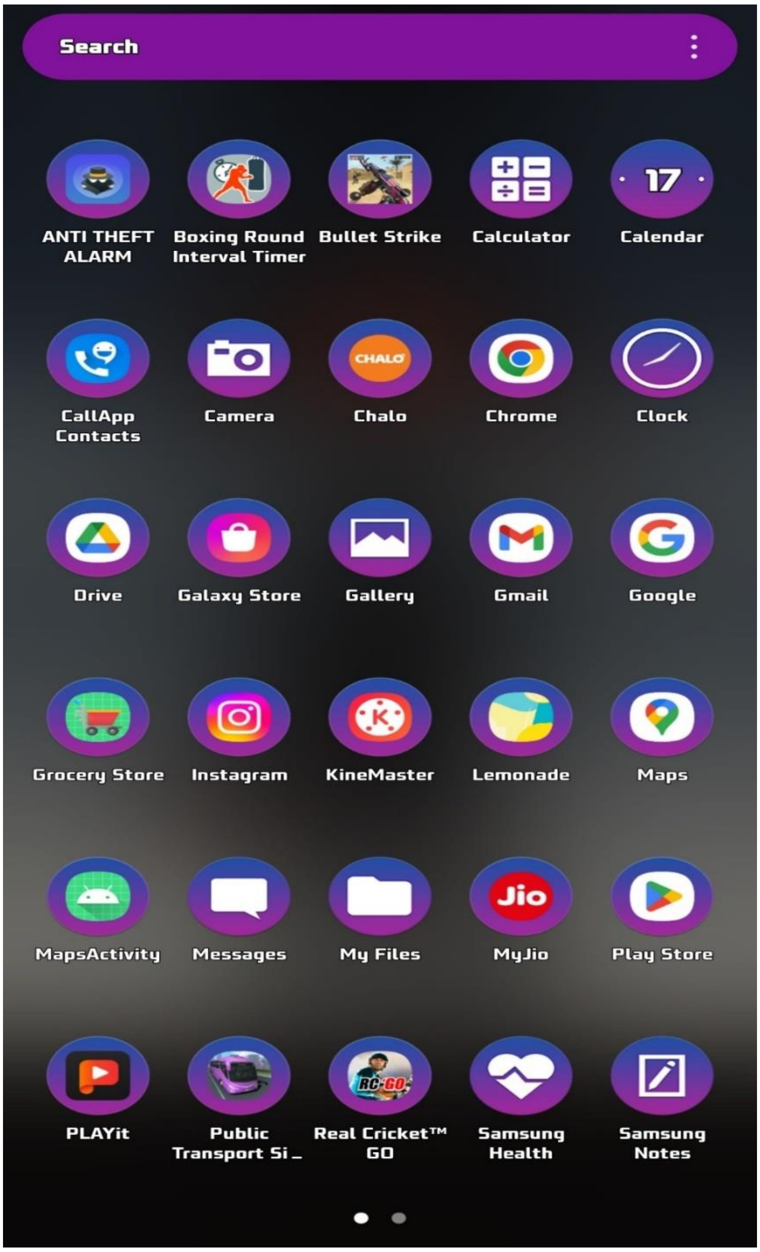


Figure 1.7: Screenshot of the Results activity

5.4 BIOMETRICS RECOGNITION

I have used the biometric technique to improve the security to the next level. I used this technique because,

Ease of transaction – It allows the user to maintain security within the transaction while eliminating the Obstructions of passwords and security questions.

Frequency of visits – This technology can increase users inclination to return.

Accessibility – Users may misplace codes and forget passwords, but they have their fingerprints on them at all times, allowing them constant access.

Fraud protection – People reuse their codes and passwords, identity-based biometrics, such as hand geometry and vein patterns, are less likely to be falsely duplicated.

Comparison and identification – Common traits and biometric information can help you better serve those who most frequent your site.

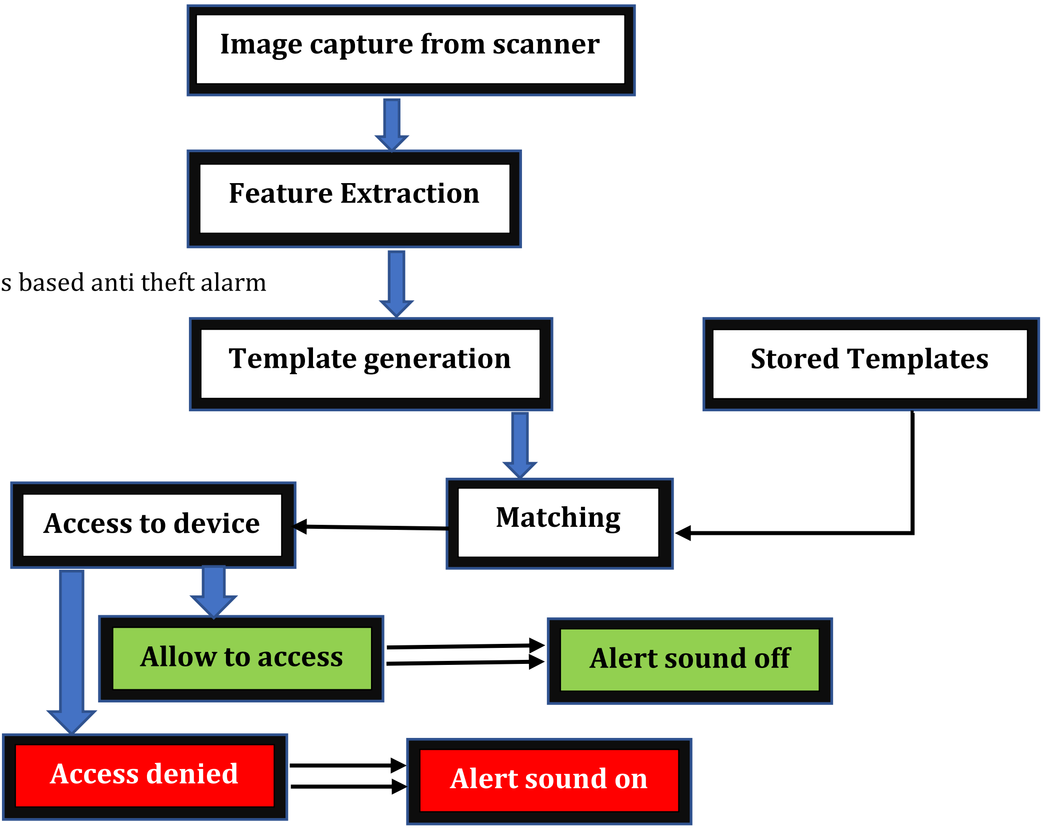
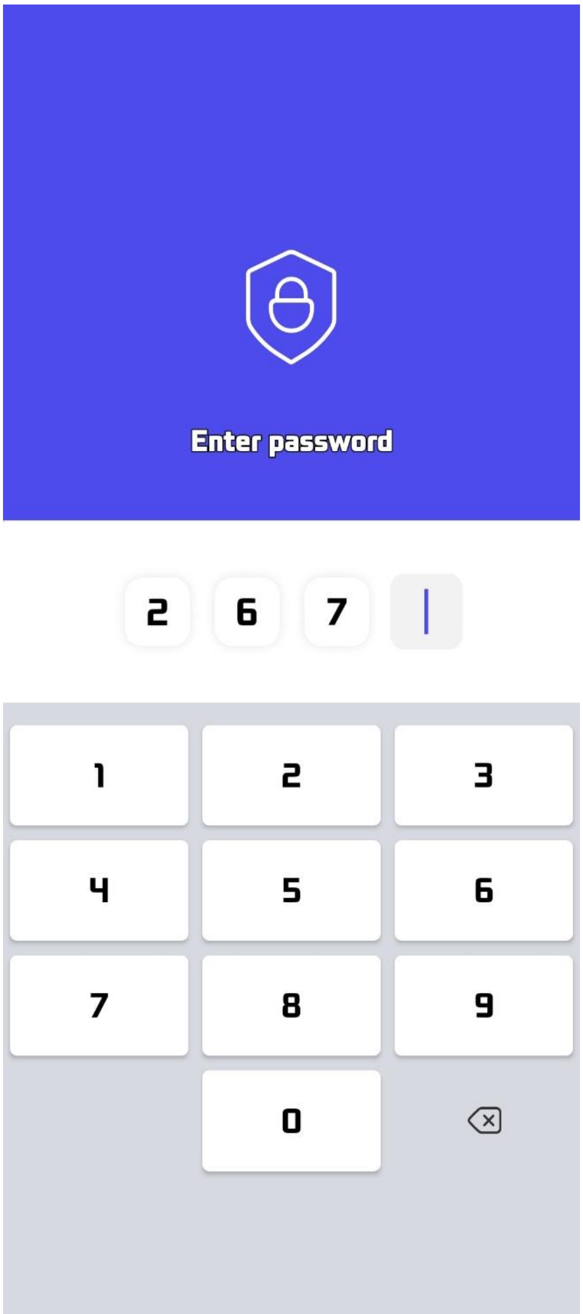
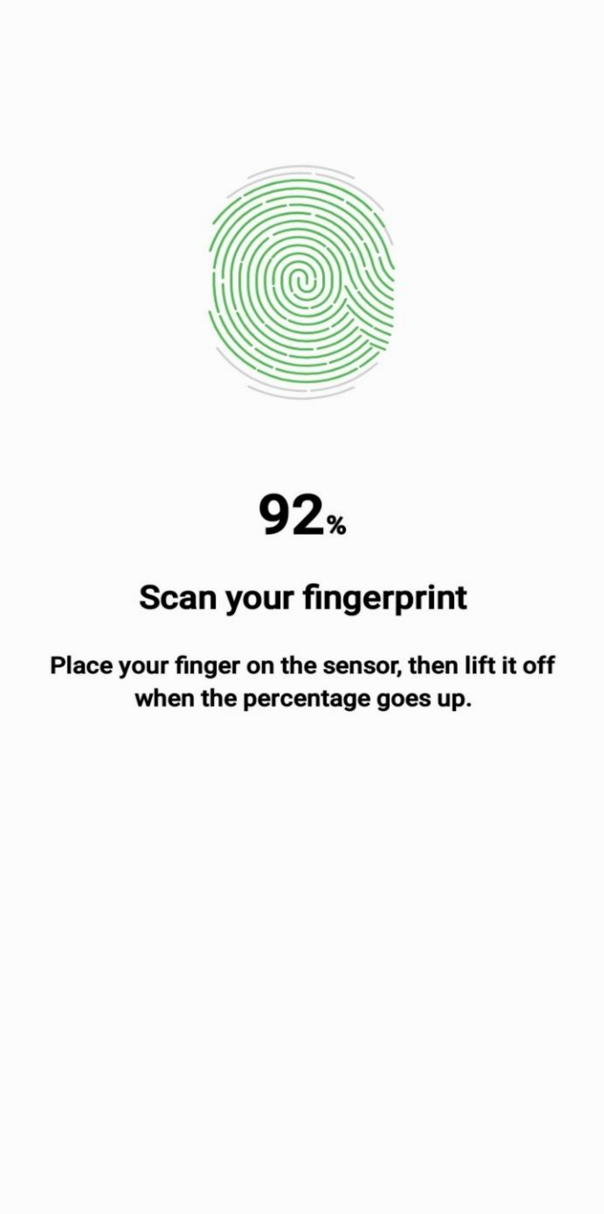


Figure 1.8.: biometrics based anti theft alarm



Enter password

Figure 1.9: Screenshot for security page



92%

Scan your fingerprint

Place your finger on the sensor, then lift it off when the percentage goes up.

Figure 2.0: Screenshot for biometric security page

5.5 ALTERNATE TECHNIQUES EXPLORED

During this project, I had been investigating multiple different techniques for each step of the project, for both the biometrics processing and the process of obtaining the real time information.

5.5.1 RECOGNITION

For recognising a anti theft alarm there was a technique is template.

TEMPLATE MATCHING

Template matching is a process where, given a sample image, it searches a second image for anything which matches this. In the case of this project, the sample image would be a sample biometrics. The idea would be that it would simply know the, if the sample image does not contain or matches the ring sound can't stop or off. If the image contains or matches at that time the alert sound can be turn off by the mobile user.

5.5.2 SETTINGS MENU

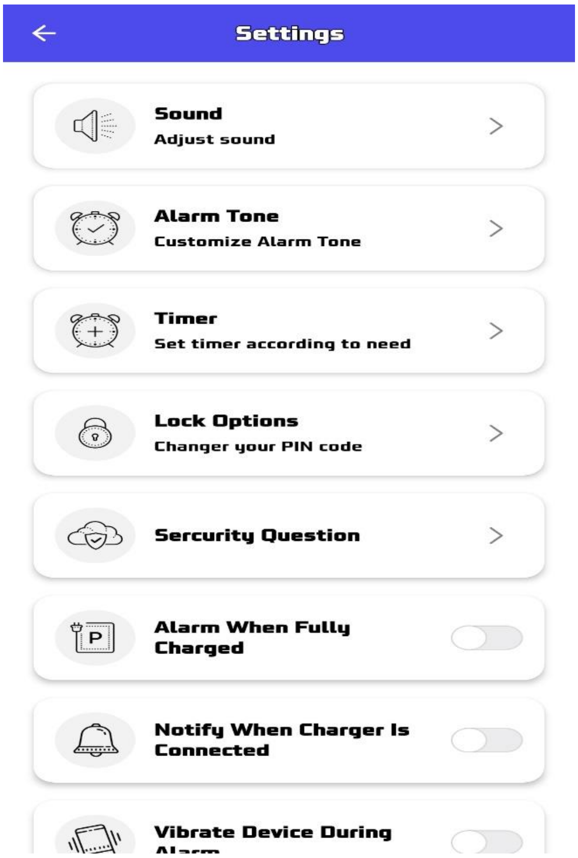


Figure 2.1: Screenshot of app menu

SOUND

In this anti theft alarm app, we can adjust the sound by increasing and decreasing the volume of the alert sound.

ALARM TONE

In this anti theft alarm app, we can change the tone of the alert sound. There is a list of tones available in it.

LOCK OPTION

In this anti theft alarm app, we can change the lock type by selecting the option. There is different lock option are

RESET PASSWORD

We reset the password. For high security and also when the user forgot his/her password

1. **Switch to pattern**

   User can change the pattern for his comfort

2. **Hide draw path**

   User can hide the drawing path so, that the intruder can't find the pattern to unlock

3. **Intruder selfie**

   If the intruder tries to unlock but he fails to unlock that that time the camera runs in background and capture the photo.

4. **Fingerprint lock**

   Fingerprint lock is a biometric lock that is the upgraded and high security level for the mobile

SECURITY QUESTION

If the user forgot the password or pattern or unable to scan the fingerprint. Instead of that this security question will ask our personal question if we answered correctly it will unlock. It I will enable once the user should give the security question before

Alarm when fully charged

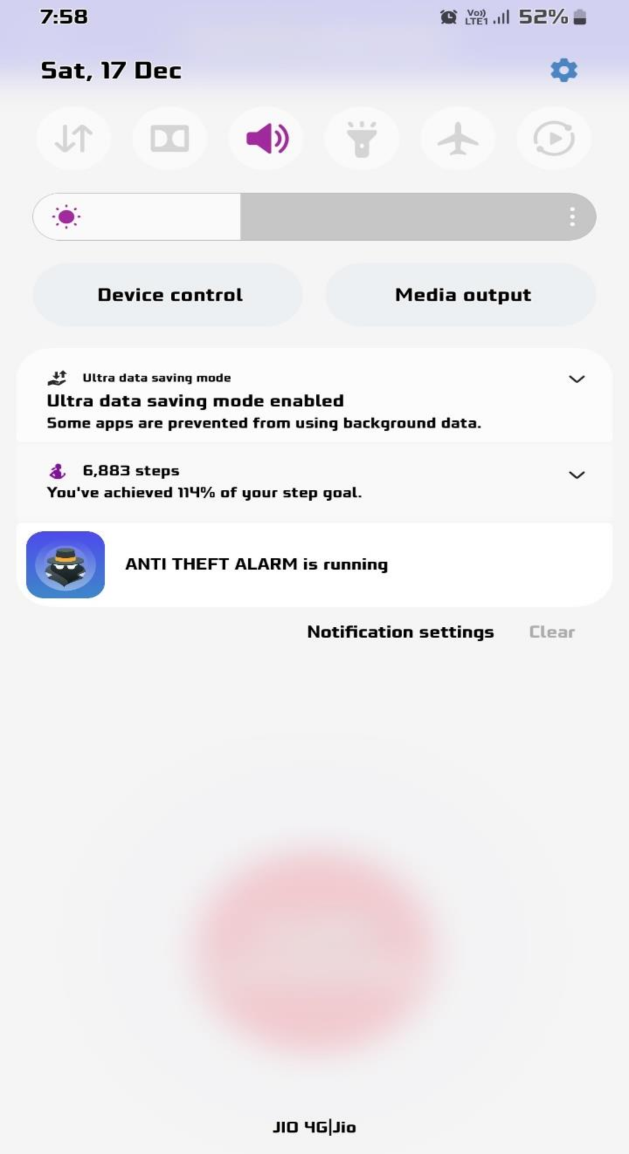
When the phone is charged fully at that time the alert sound rings so, that the user can avoid the mobile burst or damage of electrical shortage problem due to overload.

Vibrate device during alarm

When the mobile is in silent mode user can’t protect the mobile. Intruder will steal. Once the option is enabled the phone vibrate loudly so that the user will find easly.

Turn on flash light during alarm

Once option is enabled the user can find the mobile in dark room. When the intruder tries to steal the mobile flash light blinks so, that the user can spot easily even in the dark.



Ultra data saving mode
Ultra data saving mode enabled
Some apps are prevented from using background data.

6,883 steps
You've achieved 114% of your step goal.

ANTI THEFT ALARM is running

Figure 2.2: Showing the notification of app is running

FLOW CHART OF ANTI THEFT ALARM

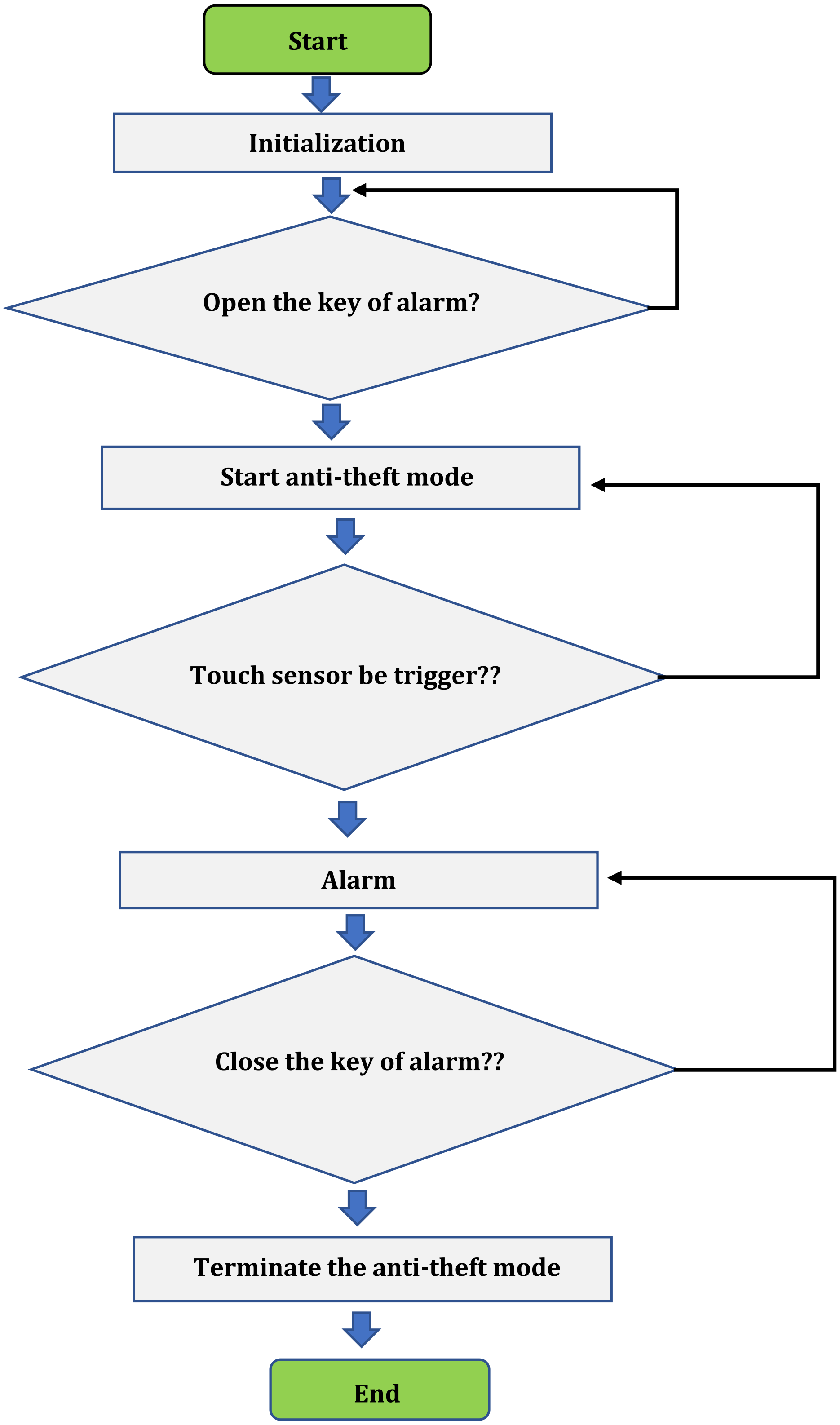


Figure 2.3 : program flow chart of anti theft alarm

5.6 CONCLUSION

In this chapter, the various methods I implemented for the development of the application were described. To start off, the application’s activities, which are the actual inner workings of the application, have been described. Following this, the computer vision techniques used and reasons for choosing them are then discussed, with images showing each step of the image processing step. Finally, there are brief descriptions on techniques which I had been investigating initially but then decided to drop from the project and reasons why.

Chapter 6

Future Work

There are many ways in which this application could be improved upon. In this chapter, I will talk about some of the features for the application which are not yet present, but could be implemented in future development.

6.1 GPS

Initially, I had been looking into using the mobile phone's GPS location retrieve the GPS position of the phone every hour with the data, longitude, latitude and location address. In that we can set the interval time to 15 minutes instead of 1 hour. User can also view in GPS location on Google Map. It retrieve position via mobile network if GPS is disabled.

Once the GPS information can be attained, the application itself could pretty much be transformed.

6.2 FINDING YOUR MOBILE WITH CONNECTED DEVICE

If you misplace your phone, your smartwatch might be able to help you find it.

To use your watch to find your phone, your phone must be:

Powered on

Connected to your watch by Bluetooth

Close enough that you can hear it

Your phone rings even if your volume is turned off and your phone is set to vibrate.

6.3 FOR IOS USER

I planned to build this app in ios platform and also i am going to upgrage this app for accessing by both android & ios users, to get beneficial, using XCODE APP DEVELOPMENT TOOL

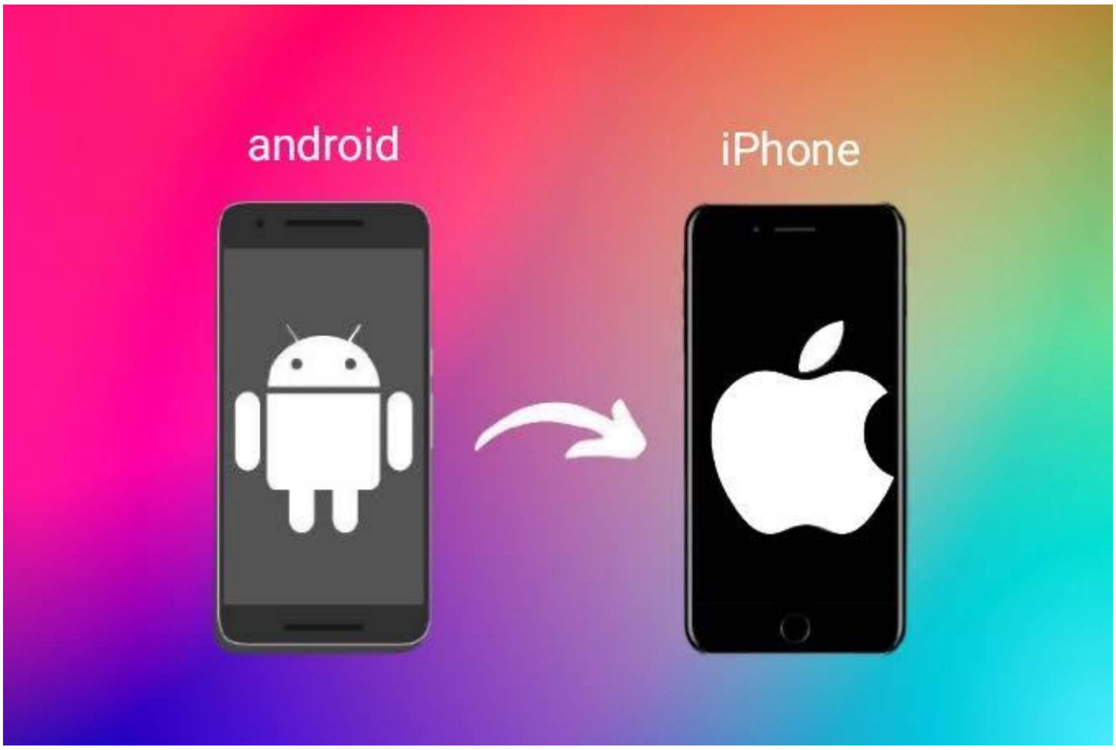


Figure 2.4: Converting the Android OS to IOS

6.4 Conclusion

In this chapter, you can see some features which could be done in future development of the application. The features are both ideas which I had originally hoped to implement 24 as part of the project from the beginning, and ideas which are just possibilities based on the current setup of the application.

CHAPTER 7

CONCLUSION

In this chapter I will talk about the results of the project. The overall aim of this project was to investigate Android App by developing an Android application which also made use of Kotlin.

The main result of the project is that there is in fact a working Android application that uses biometric and camera. A user can keep his mobile safety and they will get presented with the real time information for that mobile security.

However, it works on a very basic level. There are a few things that would need to be tidied up if the application were to be release-ready. The current state of application is only designed to work with the type of theft found in the society. The application would also need to undergo some general tidying up, both to clean up the user interface and to properly make sure the application works no matter what the external lighting conditions are.

From this project, I was able to learn about Kotlin. As I have mentioned previously, it is quite a new technology. Despite this, it has come along way in its short lifetime. Compared to the main versions of Android Studio, the Android port can be used for nearly anything. It is missing some of the lesser used functions and techniques, but it does contain all of the common functions which are used most often.