WIFI Password Management with Two Factor Authentication

MIS 3104 Individual Project

Thilina Anuradh Waasalage

A thesis presented for the Masters of Information Security

UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

UCSC
University Of Colombo
Sri Lanka
2018

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Students Name: **Thilina Anuradh Waasalage**

____

Signature: Date:

This is to certify that this thesis is based on the work of

Mr. **Thilina Anuradh Waasalage**

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

Dr. Kasun de Zoysa

____

Signature: Date:

Abstract

The major objective of this study is to implement a user friendly, feature rich mobile/web solution that supports small/corporate business employees in managing their WIFI secrets in a secure and a effective method which would make their life very easy. This solution is a breakthrough to the companies that uses the same password for all the networks also for the companies who looks to change password every week/month/once an employee leaves as the solution supports every aspects of that.

The Outcome of this project is WIFI Password Management Solution. There is no such a development done in that full fills every aspects this solution has covered.Solution is not only targeting just password management but also the security side in it.

The Solution has features like User enrollment,two factor Authentication that uses the face as the secondary authentication medium. Activity logs which will be really helpful for internal/external audits. The solution has also covered most of the possible breakouts which would enable users accessing networks with out authentication.

The implementation consists of two major developments. Android as the client and a PHP backend as the local server. Backend contains all essential informations related to WIFI and android client is a tunnel which allows the user in accessing it.

Acknowledgements

This study is the most significant and enjoyable challenge I ever had faced. With all of the obstacles I faced during the project without the great support and guidance from following people, this project would not have been completed. I pay tribute to all of them at bottom of my heart.

I would sincerely like to thank my advisor/supervisor Dr Kasun de Zoysa for his excellent guidance throughout the project.Without his exceptional generous support, encouragement this project will not be a success. Also I would also like to thank my batch mates and other friends for giving me valuable advices on the practicality and also have Providing invaluable feedback to make this project success.

Contents

List of Figures

Chapter 1

This chapter discusses the motivation behind this thesis and provides an introduction to the secure wifi password management systems with associating new technologies. Specifically this chapter provides the background of the research intent that this thesis trying to solve. It also discussed about the methodologies and procedures which will be use to implement this proposed system.

1 Introduction

The goal of this development is to provide a secure and better way of accessing WIFI related network/s in a centralized way where the user can connect to the network without the knowledge of its password and from administrators point of view a way of changing passwords and without their involment password syncronization of users with out handing over the new password unlike current manual process. Especially for secure authentication face recognition is added as the two factor verification and this would limit who ever people roaming around with un authorised access so siply this solution will not allow users to simply access even though they have credentials. Thats the basic idea of this development but before coming to that it is important to understand why we need such a solution and why this development is a breakthrough.

1.1 Motivation

Internet is an essential part of day to day life and even it has come to place where without it things won't function anymore.Which is kind of dangerous but that's the reality these days. If talked about the connectivity WIFI is the frequently used medium than generic Wired architecture due to it's ease of use and the performance. But things start getting bit complcated when so many users tries to access and we have so many things to consider like Password changes user managements security etc. In corporate environments doing these tasks is a big hassle. Even though their are solutions outheir due to their complexities & cose people tends to forget those and folow the same process and in some organiztions i have noticed they uses the same password in all the access points which is not recomended at all.So this problem indeed becomes a motifation for this whole development which targets every aspects that are essential for this problem.

1.2 Problem

We all know how to connect to a WIFI network that is easy. But if the access is given to many people, still no issue will be detected, the problem comes when the WIFI password is changed or someone in the organization wants to change the password and this is normal. In a case like this all the users who had the access should be informed manually. Not only that a person who knows the password can pass the password to another unauthorized user which is not good. So how can we manage that? So a solution will be a RADIUS based solution where the password is managed in a RADIUS server so after a proper user authentication user can access resources which are assigned to them. So the user does not have to worry about the WIFI password. Although this type of developments are in the industry such as Enterprise Wi- Fi due to its complexity those are not famous.

The negative point on them is the complexity of the development and they rely on a password or certificates we all know not completely secure.so this development will enable to integration of such password management service with face recognition module. This will give one additional layer of security. And yet this password management solution will be solely designed for android devices later improve the functionality to other devices.

Still why would anyone think of this as better solution. And people might suggest more easier ways doing this same concept rather than using a local server but then initially the internet is required at that point. We can see some good points there too but sole purpose of this development is that at working environment use these wifi accesspoints not access from anywhere concept. Still anyone could argue over this point but as we know if we keep thing in web and there are more parties involved in this but in the office environment the people who are at the office can use. Even if the users are outside it is no use unless the user is inside.So the use of a local server is indeed a good solution. Hence if someone still argues to the point where why we cannot store passwords and it is not going to change frequently. But for them yes the proper use of the development will be shown for a larger audience.Not for just 1-2 accesspoints but we could start small and clearly there is a big difference between giving a password to someone else and with out handing over the password connect automatically to it.This reduce the time as well as improves the efficiency and security also productivity with time. Just imagine a company with various kind of WIFI SSID's and when a visitor comes can they allow WIFI access to visitors.Lets say for that process you are required to register and get the credentials but still you could pass those info to someone else or when ever you want they can access giving them the full control.

So handing over the security key is rare but in this kind of scenario where no harm done on the user this scenario is very valid. So in order to secure things a two factor authentication is used. Before coming to this lets see what is meant by two factor authentication. As the word says in fact it is a way of accurately finding the validity of something. If a credit card transaction is taking place to prove that this is the users who is also the owner of this credit card. To verify we request addition information such as sending an email and if you have the access to the email meaning you can be truest. Or else sending SMS is also be possible. In fact in most of the cases we see SMS based two factor authentication.But the reason we newer used this as our second factor for authentication is that this requires money so its a burden to the user and also the code can be shared with someone easily.So as a alternative what we are trying to use instead is face recognition based two factor authentication.

Another thing we need to think in regards to WIFI is that internet connectivity prices are not cheap in srilanka. So limiting access and precautions should be taken to avoid future issues. So it is very vise to put more control on this and make this a valuable solution that assists people to manage and maintain wireless networks in a better and smarter way.

1.3 Objectives

From the development there are some features in fact the main objectives that needs to be covered that is essintial to be mentioned in this development.

The features include:

- The application should have the provision to input new WIFI SSID
- The application should have provision to assign networks to users.
- The application should have the provision to select his/her desired WIFI.
- The application should authenticate the user accurately.
- The application should stop users from playing around.

With the research(s) done a nice thread is found, in that a user is asking some interesting questions from other members on its forum.

"We are in the process of installing a new wireless infrastructure and I was curious to hear how others manage their Wifi passwords.

I will have a VLAN for wireless on the corporate lan and a VLAN for guest Wifi. For those who have a similar setup and use pre-shared keys, some questions:

1) Do you give the password for the corporate wifi out to staff? Or do you personally connect any computers that must be on the corporate wifi?

2) Do you make the guest password publicly available to staff or do you just give it out to those who ask? Or do you not give this out either, but again connect whatever machine needs guest access yourself? What about for visitors and vendors that may need guest access?

3) On a related note, do you connect your company smart phones to the corporate or guest Wifi network? I think I will be putting those devices on the guest since I don't want them scanned by my network monitoring tools like Spiceworks."[22]

So this express a need of a particular individual and with this development what we try to achieve is a solution for this issue a easy way of managing wifi passwords securely.

1.4 Scope

A fundamental goal is this development is to provice WIFI access to the users based on their authentication and their role. And also this development is targeting possible breakouts that a malicious user would try to exploit so security also integrated within this scope. To avoid password sharing a new plugable module is added as two factor authentication which would ensure that the user that tries to access is indeed the legit one. And even after a success ful attempt access is refreshed and re authentication is required again for them to re connect so this would limit un autherized access to its minimum.

The Development is consist of Input fields reporting and wifi access moules which drives this development to a success. Also this development consist of two major parts. Front end which is the client and the backend written in PHP(Laravel Framework) whch connects with an external server "KAIROS" for user(s) face authentication.
Again the Front end android consist of two sections admin/user where as the Admin has set of highlevel tasks whch only admin type users can perform and the app has the other role generic user that allows them to select his/her desired wifi SSID to connect. And if incase the user decides to generate a new password that is also possible with this login.

Backend or the local web development is only meant for administrator users where they can perform highlevel tasks like adding/managing users wifi accounts access log reads and many things. Centrl position for this whole development is the backend. Also to streanthen the security in this communication HTTPS medium is used so that any malicious attaker will not be able to perform attacks like replay etc.

From android side it is really hard to use some highlevel functions to streangthen the development as the OS itself supports only upto some level, to move further device needs to be rooted. using a rooted device just for this is not practical so all necessory measurements are placed except the things that requires root such as Device administrator which needs to be set in order to access the system if set the user is not allowed to forced close or un install this application which indeed a good step for this development.

Chapter 2

2 Review of Related Literature

2.1 Face Recognition

Face recognition is more of a topic these days even the top level social media websites like facebook instagram uses face recognision.Even though this function cannot identify users they can predict the image has a person in it. Especially if a development is done to verify a user first step will be checking the image for a face or else whats the point of further processing. So for these tasks face recoginsion simply detecting a face from an image or live videos screen. There are so many researches been conducted in this field and according to one research paper it has mensioned a large number of face recognition algorithms have been developed in last decades. These researches investigates all these methods with parameters that challenges face recognition like illumination, pose variation, facial expressions. An Interesting research paper found that discuss on the face recognition techniques with the title "A Review Paper on Face Recognition Techniques" [26]. And according to the paper wide varieties of techniques has mensioned.PCA, LDA, ICA, ANN are some of thems.

PCA (Principal Component Analysis)
Feature selection and dimension reduction is its speciality.First done by Turk and Pentland and reconstruction of human faces using PCA was done by Kirby and Sirovich.

LDA (Linear Discriminant Algorithms)
Considered as the most dominant algorithms for feature selection in appearance based methods

ICA (Independent component Analysis)
Method for finding underlying factors or components from multivariate statistical data

ANN (Artificial Neural Networks)
A new approach to face detection with Gabor wavelets & feed forward neural networks. Method used Gabor wavelet transform and feed forward points and extracting feature vectors

As this solution relys on face recognition functionity an External API was used. Google Vision API is responsible in rendering Faces from live camera and its from google.Although this API provides wide range of functionalities all what I needed was just to identify a face from a active camera image which was accomplied from this API.

2.2 Face Authentication

In theory what Face Authentication means is when a image with a face is provided identify the users. This is the basic idea.And if a social media platform like facebook is considered when a photo is uploaded they apply their face recognition algorithms and tries to find similar matches. This is something really similar to what is accomplished under this development. And for the current time most of the recent Smart phones has face authentication inbuilt. So this is not a new topic in the industry. And their are so vendors who are tring to implement a libraries to facilitate these requirements and day by day they are improving. Some of the top ranking companies that provides such servises are Microsoft,Kairos,facebook,Amazon,OpenCV etc. And even out of these only few companies are proven to provide effective results. And some provides a mechanism to integrate with lots of varieties of developments and some does not. And each solution has a complete different algorithm and different set of specialities.

If a full scenario is considered how this process is done in multiple steps this whole process can be acomplished.
Each face algorithem gives a marging to accept and the accuracy of this process is depends on that threashold.

- Generate a FaceID and store with a unque name

When generating this unque ID all the face points are convered and a face map is generated and saved in a database.

- On Recognition a faceID is generated adn compared with the database and order based on the acuracy level 0-1

2.2.1 Kairos

Based on the researchs done Kairos is the only company that provides a overall product that covers most of the importent aspects juch as Face Detection,Recognition,Video face recognition,SDK to work offline & API.

2.2.2 Amazon Rekognition

Amazon Rekognition also provides highly accurate facial analysis and facial recognition on images and video that a user provide. A user can detect, analyze, and compare faces for a wide variety of user verification, people counting, and public safety use cases.

2.3 Face ID Spoofing

After analyzing reviews and weeknesses that could go wrong is face based authentication this topic is at the top. Meaning a face that looks similer to the original user or a photo captured from the original user used against the system to by pass the authentication process.

Types of attacks:

- Attack using a photo of the original user
- Attack using a video that records of the original user
- Attack using a face mask that has the exact same features like the original users face

2.4 Face ID Spoof Detection

Spoof detection simply stands for mechanisms that detect false attempts to by pass the face authentication. A valid idea behind this detection would be to detect if any liveness available in this detection. If any found the authentication process is valid. If an image is shown instead there will be liveness in the image but if a video is considered this logic cannot be considered.

Through the investigations done blinking is said to me a good detection mechnism where as the user needs to blink its eye to provide a command to perform the checking. Although this will not be working for videos but this will not let photos to fool the solution. Also their is one thing left and it is bit complicated what it detects is the background facts like frames and stuff. If any suspicious item is detected will not authorised to proceed. If an attacker uses image as this authentication rather than considering just face read background items carefully and comes to a conclusion.Although this is bit more theoritical in practice this will not be used as this process may lead to lower performance. If the time is not a valid factor but accuracy and security is then this process is recomended.

2.5 WIFI Auto connection

This was the next big step once the authentication is completed user is required to get connected to defined SSID with the password. But this password should not be revealed to the user at any cost. So as a research I was able to connect to 2-3 wifi SSIDs with different passwords (Hardcoded for testing purposes) and re-create the practical scenario. With this without any user interaction I was able to connect to WIFI SSID from android device even if the WIFI module is switched off. So with this research section what I tried was to enable wifi module if disabled and connect with a wifi network without users interation.

2.6 Fake WIFI networks

An Evil Twin hotspot is a Wi-Fi access point set up by a hacker or cybercriminal. It mimics a legitimate hotspot, including the service set identifier (SSID), also known as the primary network name, provided by a nearby business, such as a coffee shop that provides free Wi-Fi access to its patrons.

If an attacker tried to re create a SSID without any password and let a user logs in to this via the application the user will think he's connected to legit wifi but its not.This would only happen if the hackers wifi signal streanth is far supirior than that of the original access point. Although this is a serious security breach this cannot be stoped.

2.7 Related Research works

2.7.1 ENTERPRISE Wi-Fi solution from Xirrus

Its a company that supercharges a business with their robust Wi-Fi solutions that are secure, simple to operate. And according to them they provide, "INSTANT WI-FI ACCESS WITH EASYPASS.".

Security and simplicity and provides guests, visitors, and BYOD users quick access to the Wi-Fi network. IT administrators have the ability to track user activities and protect data. [3]

2.7.2 Hardware Device for Authentication from IriShield

I have hands on experience with this device for close to 1 year so I know exactly how this works and yes it provides accurate results most of the times. With the Android, Windows SDK they have currently you can easily integrate it with RADIUS server, but the problem lies in the cost, price on this device is high. And this device contains user informations as templates, and all the users in the system can be enrolled (Registered manually) and can be validated. [4]

2.7.3 RADIUS Server

Radius is typically used as a authentication method to control who can get access to a device, or who can connect using a VPN client.This process in highly used in corporate environment. So in this case if the user could proves he is legitimate and one the RADIUS server validates the user access is given to the user. This is done commonly done with passwords and digital certificates.

2.7.4 WPA2-Enterprise

"WPA2-Enterprise with 802.1x authentication can be used to authenticate users or computers in a domain. The supplicant (wireless client) authenticates against the RADIUS server (authentication server) using an EAP method configured on the RADIUS server. The gateway APs (authenticator) role is to send authentication messages between the supplicant and authentication server. This means the RADIUS server is responsible for authenticating users.

APs perform EAPOL exchanges between the supplicant and convert these to RADIUS Access-requests messages, which are sent to the RADIUS server's IP address and UDP port specified in Dashboard. Gateway APs need to receive a RADIUS Access-accept message from the RADIUS server in order to grant the supplicant access to the network.

For best performance, it is recommended to have the RADIUS server and gateway APs located within the same layer-2 broadcast domain to avoid firewall, routing, or authentication delays. Keep in mind the AP is not responsible for authenticating wireless clients and acts as an intermediary between clients and the RADIUS server."[23]

The below diagram shows how this is done.

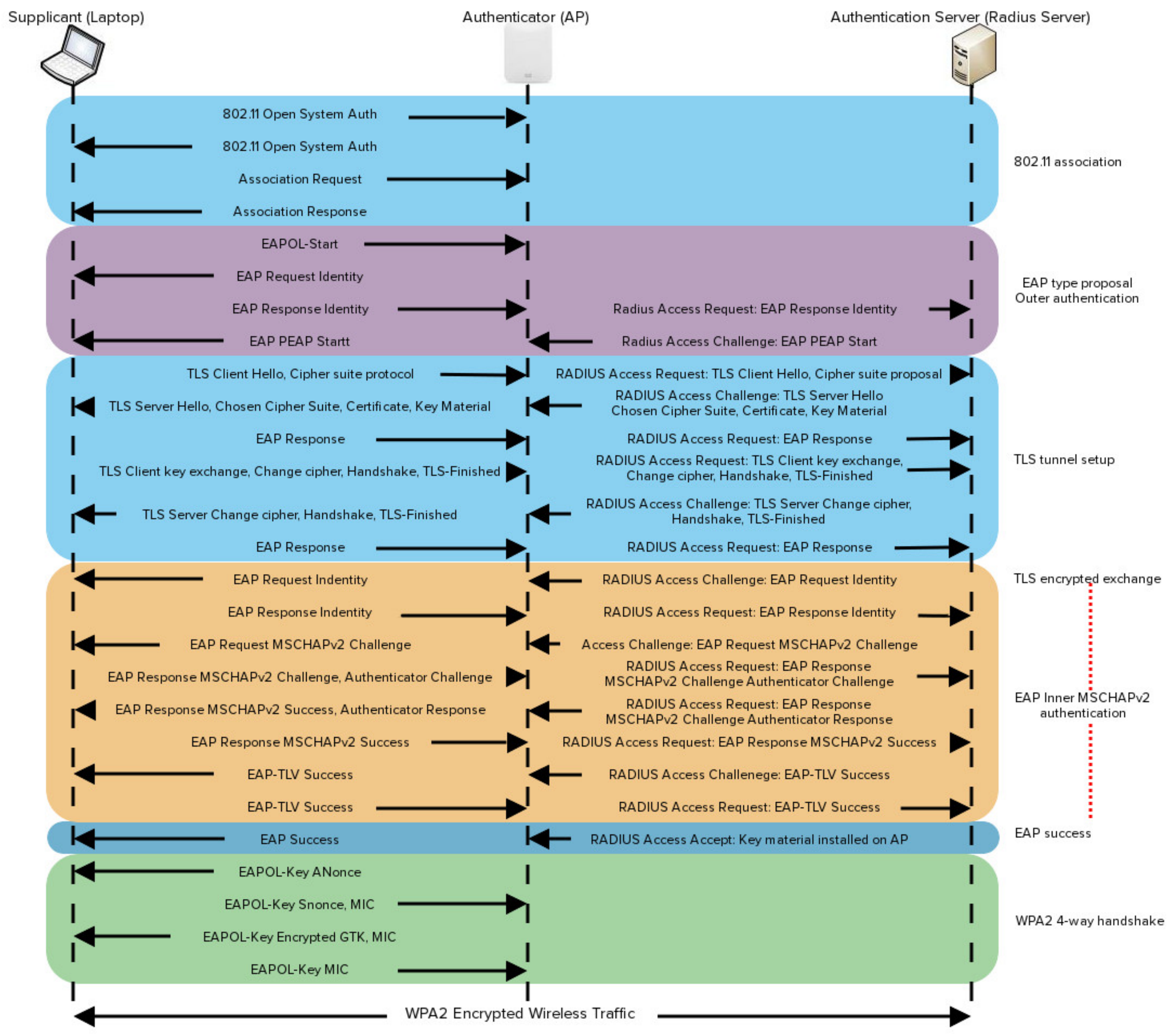

Supplicant (Laptop)
Authenticator (AP)
Authentication Server (Radius Server)
802.11 Open System Auth
802.11 Open System Auth
Association Request
Association Response
802.11 association
EAPOL-Start
EAP Request Identity
EAP Response Identity
Radius Access Request: EAP Response Identity
EAP PEAP Startt
Radius Access Challenge: EAP PEAP Start
EAP type proposal
Outer authentication
TLS Client Hello, Cipher suite protocol
RADIUS Access Request: TLS Client Hello, Cipher suite proposal
TLS Server Hello, Chosen Cipher Suite, Certificate, Key Material
RADIUS Access Challenge: TLS Server Hello
Chosen Cipher Suite, Certificate, Key Material
EAP Response
RADIUS Access Request: EAP Response
TLS Client key exchange, Change cipher, Handshake, TLS-Finished
RADIUS Access Request: TLS Client key exchange,
Change cipher, Handshake, TLS-Finished
TLS tunnel setup
TLS Server Change cipher, Handshake, TLS-Finished
RADIUS Access Challenge: TLS Server Change cipher,
Handshake, TLS-Finished
EAP Response
RADIUS Access Request: EAP Response
TLS encrypted exchange
EAP Request Indentity
RADIUS Access Challenge: EAP Request Identity
EAP Response Indentity
RADIUS Access Request: EAP Response Identity
EAP Request MSCHAPv2 Challenge
Access Challenge: EAP Request MSCHAPv2 Challenge
EAP Response MSCHAPv2 Challenge, Authenticator Challenge
RADIUS Access Request: EAP Response
MSCHAPv2 Challenge Authenticator Challenge
EAP Response MSCHAPv2 Success, Authenticator Response
RADIUS Access Request: EAP Response
MSCHAPv2 Challenge Authenticator Response
EAP Inner MSCHAPv2
authentication
EAP Response MSCHAPv2 Success
RADIUS Access Request: EAP Response MSCHAPv2 Success
EAP-TLV Success
RADIUS Access Challenege: EAP-TLV Success
EAP-TLV Success
RADIUS Access Request: EAP-TLV Success
EAP Success
RADIUS Access Accept: Key material installed on AP
EAP success
EAPOL-Key ANonce
EAPOL-Key Snonce, MIC
EAPOL-Key Encrypted GTK, MIC
EAPOL-Key MIC
WPA2 4-way handshake
WPA2 Encrypted Wireless Traffic

Figure 1: Detailed breakdown of the PEAP with MSCHAPv2 [23]

2.8 Android System Overview

Android is a software stack that includes a Linux based Operating System, middleware and key application. It is designed to run on touch screen mobile devices like smart phones and tablet computers. Android Inc. was founded in Palo Alto of California,

U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later, Android Inc. was acquired by Google in 2005. It is now owned by Google in conjunction with Open Handset Alliance. This Alliance is a collection of 86 hardware, software and telecommunication companies developing open standards for mobile devices.

Applications in Android are typically developed in Java programming language using the Android Software Development Kit (SDK). The java code is compiled into byte code (.class files) that is platform independent. The Dalvik interpreter converts the Java Virtual Machine compatible .class files into Dalvik Executable files (.dex files) for runtime interpretation.

The Android SDK provides the necessary Application Programming Interface (APIs) and developer tools for building, testing and debugging applications in Android.

An Android application consists of one or more of the following components: Activities,Services, Content Providers and Broadcast Receivers. Each component has a specific role and contributes to the overall application behavior. All the components used in the application and the device features are declared in the Manifest file. The Android application is also composed of Resources that are separate from the source code. These resources like the images, audio files, etc. contribute to the visual presentation of the application.

In the subsequent sections, we will discuss in detail about the architecture, features and application framework of the android operating system.

2.9 Android Architecture

Android Operating System (OS) can be considered as a software stack consisting of various layers, where each layer is a group of several different components. The layers are:

- Applications
- Framework Services and Libraries
- Native Libraries, Daemons and Services
- The Linux Kernel which includes drivers for hardware, networking, file system access and inter-process-communication.

Each layer in the architecture provides different services to the layer above it. Above figure shows the overview of Android Architecture. In the following sections, we will discuss each layer in detail.

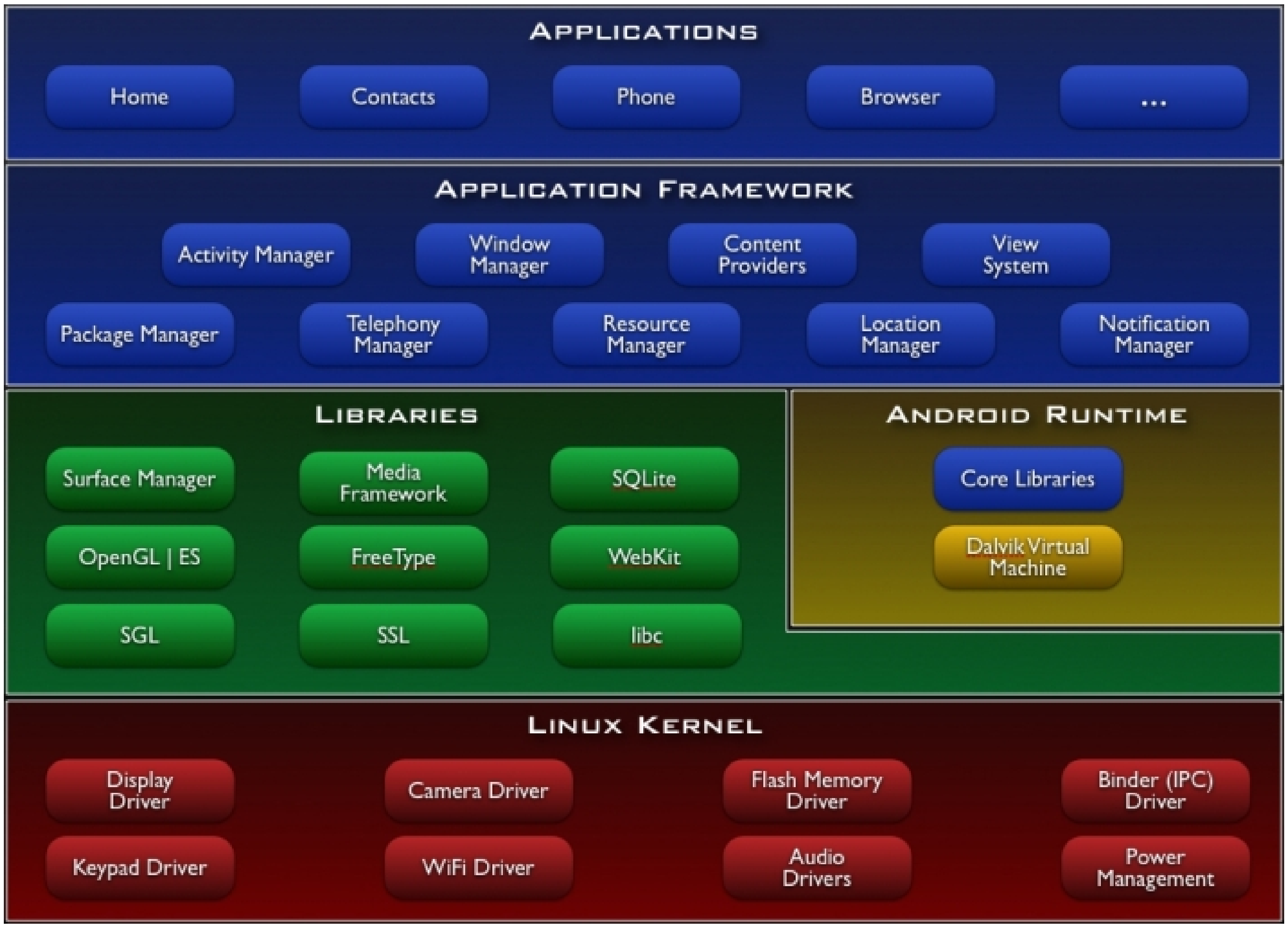


Figure 2: Overview of Android architecture [16]

2.9.1 Application Layer

Applications form the top layer of the Android Architecture. Android provides a set of core applications such as: Email Client, SMS Program, Web Browser, Calendar,Contact Manager,Maps etc. User defined applications can also interact with these core applications.

All applications are written in Java programming language. The Android SDK tools compile all data and resources into an Android package which is a file with .apk extension. This file is considered as one application and is used to install the application on any Android device.

Once installed on a device, the Android OS treats each application as a different user thus making the Android OS a multi-user Linux system. The system assigns a user ID to the application, sets permissions to all the files that the application needs so that only the user ID assigned to the application can access them. Every application runs in its own Linux process, each process has its own virtual machine, so an applications code runs independently from other applications.

By default, each application only has access to its own components and cannot access any part of the system without permission, thus creating a very secure environment. However, it is possible for an application to access other applications and system services. Two applications share information between them by sharing the same user ID. An application can request permission to access devices data like contacts, camera etc. which will be granted at the application install time. Android applications consist of one or more application components [3].

Application Components form the essential building blocks of an application and each one is distinct in the way it contributes to the overall behavior of the application. There are four types of components. They are:

- **Activities:** An Activity is a screen that the user sees and interacts within an application.Each activity runs independently and one application can start an activity of another application.
- **Services:** This is a component that does not have a user interface. It runs in the background to perform long-running operations or to work on remote processes. An example of service would be to play music in the background while the user is accessing a different application.
- **Content Providers:** This component provides a way to share data across applications. Using content providers, data can be retrieved, created and modified in the application.
- **Broadcast Receivers:** This component responds to any broadcast announcements.This is one of the well used feature in android SDK. Many broadcast are made by the system, but can also be created and started by the application. They are intended to do very little work and are therefore considered as gateways to other components.Few example for this topic would be Trigger is a SMS comes, trigger if a call comes etc.

2.9.2 Application Framework Layer

Application Framework is a layer that the Application Layer directly interacts with. It is a toolkit that enables the reuse of components: an application can expose its capabilities and any other application may then make use of those capabilities. This layer however enforces security constraints on the capabilities. Developers thus have full access to these components and can use them or replace them and implement them in their own way.

Important Components of the Application Framework that were used in the application are:

- **Activity Manager:** Manages the lifecycle of applications and provides a common navigation backstack.
- **Content Providers:** Enables applications to access data from other applications or to share their own data.
- **Resource Manager:** Manages the resources that are not part of the code like graphics, image files etc.
- **View System:** Helps in building of an application, including text boxes, buttons etc.

2.9.3 Application Libraries

The layer below the Android Application Framework is the Androids native libraries. This layer exposes the different capabilities through the Application Framework. This layer can be treated as a set of instructions that enables the device to handle different kinds of data. For example, the playback and recording of any audio, video, picture formats is supported by the media framework library.

These libraries are written in C or C++ programming language depending on the hardware being used.
Some of the important native libraries are:

- **Surface Manager:** The Surface manager handles the off-screen buffering for the window manager from the framework layer. Off-screen buffering implies that drawings cannot be drawn directly onto the screen but are put into an off-screen buffer. There it is combined with other drawings to form the final screen which the user can see. The transparency of windows is achieved because of this off-screen buffer.
- **OpenGL:** The GL stands for Graphic Library. It is mainly used for rendering the content of 2D or 3D graphics to the screen.
- **Media Framework:** This library supports different media codecs allowing the recording and playback of various media formats (audio, video, picture, etc.)
- **FreeType:** It is mainly used for rendering of bitmap and vector font.
- **SSL:** SSL stands for Secure Sockets Layer. It is a cryptographic protocol used for secure communication over the internet.
- **SQLite:** It is lightweight relational database engine used for the data storage purposes in android.
- **WebKit:** This library provides tools for browsing the web pages. It has a HTML renderer, a JavaScript Engine and a cookie manager. These tools facilitate the displaying of any web content inside an application.
- **Libc:** It is a standard C Library tuned for embedded Linux based devices.

2.9.4 Android Runtime

The Android Runtime is located in the same layer as the native libraries. It consists of the Dalvik Virtual Machine and the Core Java Libraries.

2.9.4.1 Dalvik Virtual Machine

Android uses its own virtual machine which is called as the Dalvik Virtual Machine. This is an interpreter and was developed by Dan Bornstein of Google. The Dalvik Virtual Machine (VM) is not compatible with the standard Java Virtual Machine as it is specialized and optimized for small systems. These small systems usually have low processing power and low memory.

Java code is compiled by the java compiler and generates the .class files. In Android, these .class files are converted to .dex files (dalvik executable format) at the time of compilation and are then interpreted by the Dalvik VM.

The Dalvik VM supports multiple virtual machine processes per device thus providing high efficiency in low resource environments. The Dalvik VM through the creation of multiple processes also ensures security, isolation of one application from another.

2.9.4.2 ART

The Android RunTime, ART, is the successor and replacement for Dalvik, the virtual machine on which Android Java code is executed on.

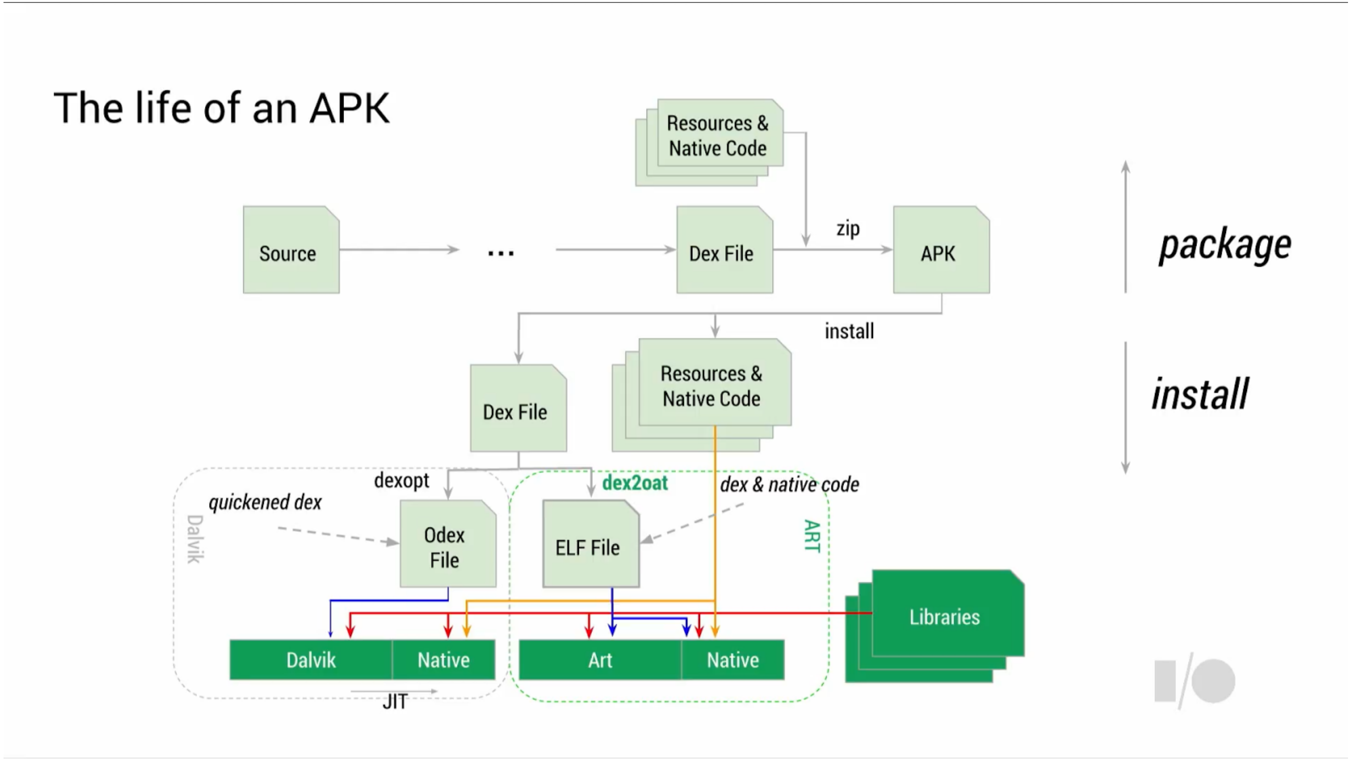


Figure 3: The Life of an APK [17]

The big paradigm-shift that ART brings, is that instead of being a Just-in-Time (JIT) compiler, it now compiles application code Ahead-of-Time (AOT). The runtime goes from having to compile from bytecode to native code each time you run an application, to having it to do it only once, and any subsequent execution from that point forward is done from the existing compiled native code.

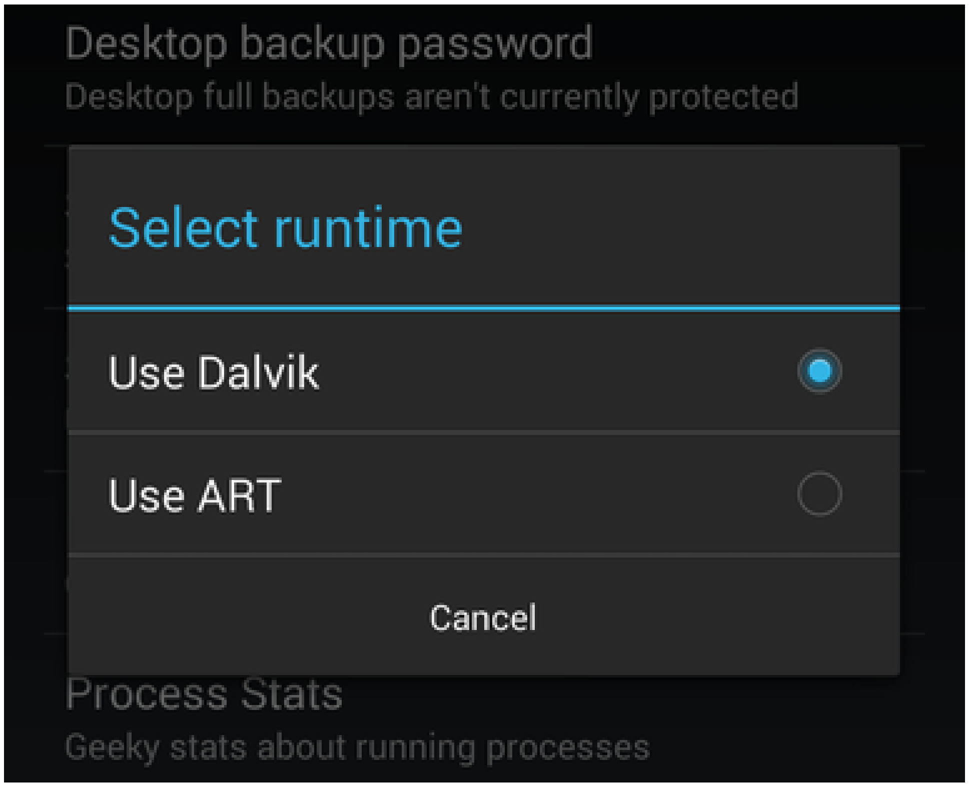
2.9.4.3 ART vs Dalvik

Dalvik is based on JIT (just in time) compilation. It means that each time you run an app, the part of the code required for its execution is going to be translated (compiled) to machine code at that moment. As you progress through the app, additional code is going to be compiled and cached, so that the system can reuse the code while the app is running.

Since JIT compiles only a part of the code, it has a smaller memory footprint and uses less physical space on the device.

ART, on the other hand, compiles the intermediate language, Dalvik bytecode, into a system-dependent binary. The whole code of the app will be pre-compiled during install (once), thus removing the lag that we see when we open an app on our device. With no need for JIT compilation, the code should execute much faster.

You can choose to use it by going to "Settings— Developer Options— Select Runtime" and choosing ART.



Desktop backup password

Desktop full backups aren't currently protected

Select runtime

[✓ Use Dalvik]

Use ART

Cancel

Process Stats

Geeky stats about running processes

Figure 4: Choosing AR or Dalvik [18]

2.9.4.4 Core Java Libbraries

Android applications are written in Java Programming Language and with the introduction of Kotlin in Google I/O 2017 more developers tends to use Kotlin as well as it works perfectly in JRE and the coding is similar to IOS and web development. And in regards to these libraries provide most of the functionality necessary for building the applications. Some of these functionalities include data structures, file access, network access, graphic support, etc.

2.9.5 The Linux Kernel Layer

The basic layer on which all other layers are built is the Linux Kernel layer. It uses the Linux 2.6 Operating System to perform all of the core system services like process management, memory management, networking, security settings, file management, etc. It is this layer that interacts with the hardware and includes all of the essential hardware drivers.

Basically what you see on the android device is that it runs a linux kernal and top level coding is build on top of it to make things easy. Consider an example where we require to capture the current battery percentage we can do this with few android lines. But how this actually works is that all the percentages are saved in a text file.So from linux kernal that file is written and updated and then shared with the Android API.

Drivers are software programs that control and communicate with the hardware devices. For example, for any device that has Bluetooth hardware in it, the kernel would contain a Bluetooth driver that communicates with the Bluetooth driver. This layer also acts as an abstraction layer between the hardware and the rest of the software layers.

2.10 Android Platform Features

There are several versions that have been released for Android Platform. Each version added some new features and fixed any bugs found in the older versions. Some of the key features of Android include:

- **Rich Multimedia:** Android supports a wide variety of common media types. Audio, video, images can easily be integrated into an application with the multimedia framework which is supported by Android. The Android Framework also supports various cameras and their features available on devices, allowing user to capture pictures and videos for any application.

- **Location and Maps:** The Android location framework can be used to acquire the user location which can be tracked and the location services can also be used by other applications. This framework does not rely on GPS for identifying the user location; the device therefore consumes less battery. The Google Maps External Library provides support for displaying and managing Maps data.

- **Animation and Graphics:** Android Animation can be applied to enhance the look of the User Interface (UI). Android includes support for high performance 2D and 3D graphics with the Open Graphics Library (OpenGL).

- **Web Browser Support:** The Android platform supports a web browser which is based on open source WebKit application framework and is collaborated with Google Chromes V8 JavaScript engine. The web browser also provides support for HTML5.

- **Data Storage:** Android provides several storage options for the user to save application data persistently. Data can be stored in SQLite which stores structured data in a database, or in SharedPreferences, which stores private primitive data in key-value pairs, internal storage which uses the device memory or removable storage.

- **NDK:** NDK stands for Native Development Kit. It is a toolset that allows developers to implement parts of the application using native code languages like C and C++. This toolset will be helpful because developers can reuse any existing code libraries written in these languages

2.11 Android Applications

Android Applications form the highest layer in the Android Architecture. This is implemented in Java Programming Language. An android application consists of one of more of the application components: Activities,Services, Broadcast Receivers and Content Providers. The first three components are activated by an asynchronous message called as Intent. Intents bind individual components at runtime.

An android application also contains a manifest file that declares all the components being used in the application along with the application requirements like the hardware configurations required, minimum version of android required, etc. An android application also consists on application resources like images, strings, layout files, etc., which is not part of the code but contributes to the various aspects of the application. Each of these is discussed in detail in the following sections.

2.12 Application Components

The most essential building blocks of an Android application are the application components. Each component plays a specific role in defining the applications overall behavior. The four different types of application components are:

2.12.1 Activities

An activity is an application component that provides a screen (view) with which a user interacts and does some work, like sending an email, taking a photo, etc. Each activity is given a window where the user interface is drawn. Activity therefore, forms the building block of the user interface. In the Model View Controller (MVC) Architecture, the Activity forms the Controller and handles all the events related to the view. An application usually consists of multiple activities that are bound together. Each activity communicates with other activities to send and/or retrieve data.

2.12.2 Content Providers

Content Providers are mainly used to manage access to structured set of data. The data can be a SQLite database, or a file system or any other storage location that the application has access to. Content Providers encapsulate the data and provide mechanisms for securing the data. They form the primary interface for connecting data in one process with code running in another process. Using the Content Provider, other applications can query or modify (or do both) data. For example, Android maintains a content provider for the contact information of the user. So, any application that has proper permission can query the information of any particular person from the contacts. Content Providers can also be used to write and read data that is private to an application and not shared.

2.12.3 Services

A service is an application component that runs in the background and does not interact with the user interface. It is usually used for performing long running operations such as playing music in the background while the user is in another application, bringing data over the network without blocking user interaction in an application, perform file I/O, or interact with the Content Provider, etc.

2.12.4 The Manifest File

Every application in Android has a manifest file and we consider this file to be the heart of the application as by this looking at it we could get a clear picture of how the application looks like. This is named in the application as AndroidManifest.xml. This file presents essential information about the application to the Android system. All the components which the application would require to perform its work will be defined in this manifest file. The system reads this file to check if the application component exists before starting it. The manifest file is kept in the root of the applications project directory. The android manifest file also contains information about:

- **Permissions:** It declares the permissions that the application has in order to access any protected parts of the API, or to interact with other applications. For example, it identifies the user permission that the application requires for Internet access, or read only access to users contacts, etc. It also declares the permissions that other applications require in order to access the components of the application for which the manifest file is written.

- **Package:** Names the java package for the application, the package name becomes the unique identifier for the application.

- **Application Components:** Declares and describes the components of the application (activities, services, broadcast receivers, and content providers). These declarations let the Android system know which components can be used by the application and the circumstances in which they can be launched.

- **Minimum API level:** Depending on which APIs the application uses, it declares the minimum API level required by the application.

- **Libraries:** Any libraries other than the Android Framework APIs are also listed in the manifest file. For example, Google Maps Library if being used by the application must be declared in the manifest file.

- **Application Features:** Declares any hardware and software features that the application needs, like Bluetooth services, camera, multi-touch screen, etc.

2.12.5 Application Resources

Application Resources are part of the android application but are not part of the application code. They contribute to the visual presentation of the application. Some of the resources include images, audio files, menus, styles, colors, animation, layout of an activity, etc. Some of these resources are defined in XML files.

Application Resources are defined /res folder of the root directory. The /res folder contains various directories for resources. For example, the image logo.png is usually placed in the /res/drawable/ directory. Every resource is associated with a unique integer ID (defined by the SDK build tools) and this ID can be used to reference the resource either from application code or from other resources defined in the XML files. These resource IDs are defined in your applications R class and are generated automatically. The logo.png can be accessed using R.drawable.logo in application code or @drawable/logo in XML.

Android provides a set of alternative resources by specifying qualifiers for the alternative resources. A qualifier is a short string that is included in the name of the resources directory. They help in defining the device configurations for which the resources should be used. For example, if the UI strings in XML are to be supported in both English and French languages, the strings can translated into French and stored in separate files. The directory for storing the English XML would be /res/values and the directory for French XML would be /res/values-fr/. Here the fr becomes the qualifier. Then based on the language setting selected by the user, the appropriate language strings get applied to the UI.

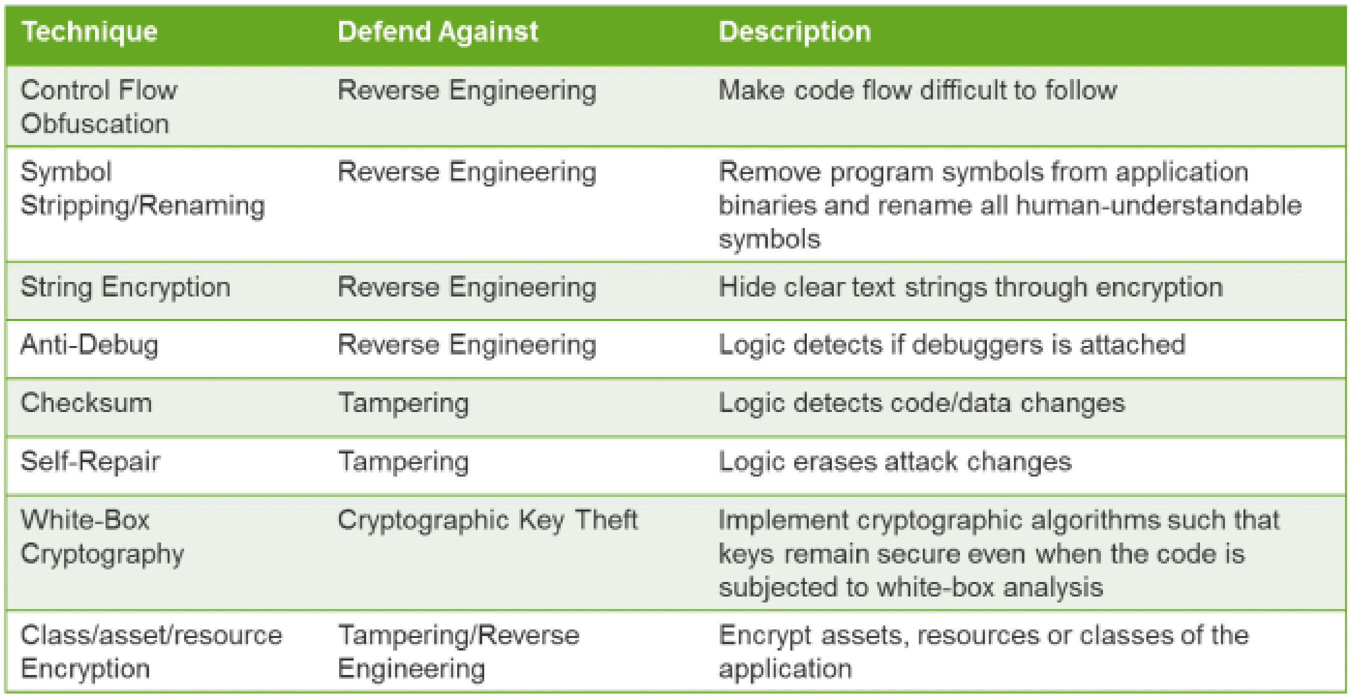
The XMLs for supporting the UI in the portrait mode are defined in the /res/layout/ directory; the XMLs for supporting the UI in the landscape mode are defined in the /res/layout-land/ directory.

2.12.6 Android Security

Android is open source and we can find plenty of os source code if we do a small research and you can do magic with android like things you cannot do with IOS and that is one of the main reason that people tends to use android over other platform. As they feel it so easy to use applications and there are plenti of android application and from developer point of view each year a complete new android SDK comes in to the market with whole bunch of new API which allows developers to build app to the current trend with limited efforts. So with these upgrades someof the functionalities used previously might get depriciated and then could cause a weakness of the application it self. And this opens up new threats and also if the device is rooted the developer could do things to harm the user.

With root permisions you could turn the whole os up side down like restarting your phone access your sensitive data with our your approval and as the ultimate last result could destroy your phone via restarting on boot up. And we have plenty of android applications in google play and other markets too. But if we talk about the google market they don't check application functionalities they just check the screenshots and descriptions using some robots so as a result app publication is easy but we cannot trust. But in IOS this is not the case each app is verified through a manual process that could take weeks which gurantees the app harmfullness to a minimum.Also in terms of developers perspectives a malicious users could alter the application and try making the application perform bad intents. So the user could crack and change the way app behaves by changing the source code. This is what we have to stop. But this is something we cannot avoid fully but we could limit its harm.

2.12.7 Ways of Protecting client-side code



Technique	Defend Against	Description
Control Flow Obfuscation	Reverse Engineering	Make code flow difficult to follow
Symbol Stripping/Renaming	Reverse Engineering	Remove program symbols from application binaries and rename all human-understandable symbols
String Encryption	Reverse Engineering	Hide clear text strings through encryption
Anti-Debug	Reverse Engineering	Logic detects if debuggers is attached
Checksum	Tampering	Logic detects code/data changes
Self-Repair	Tampering	Logic erases attack changes
White-Box Cryptography	Cryptographic Key Theft	Implement cryptographic algorithms such that keys remain secure even when the code is subjected to white-box analysis
Class/asset/resource Encryption	Tampering/Reverse Engineering	Encrypt assets, resources or classes of the application

Figure 5: Client side code security [19]

2.13 Kairos API

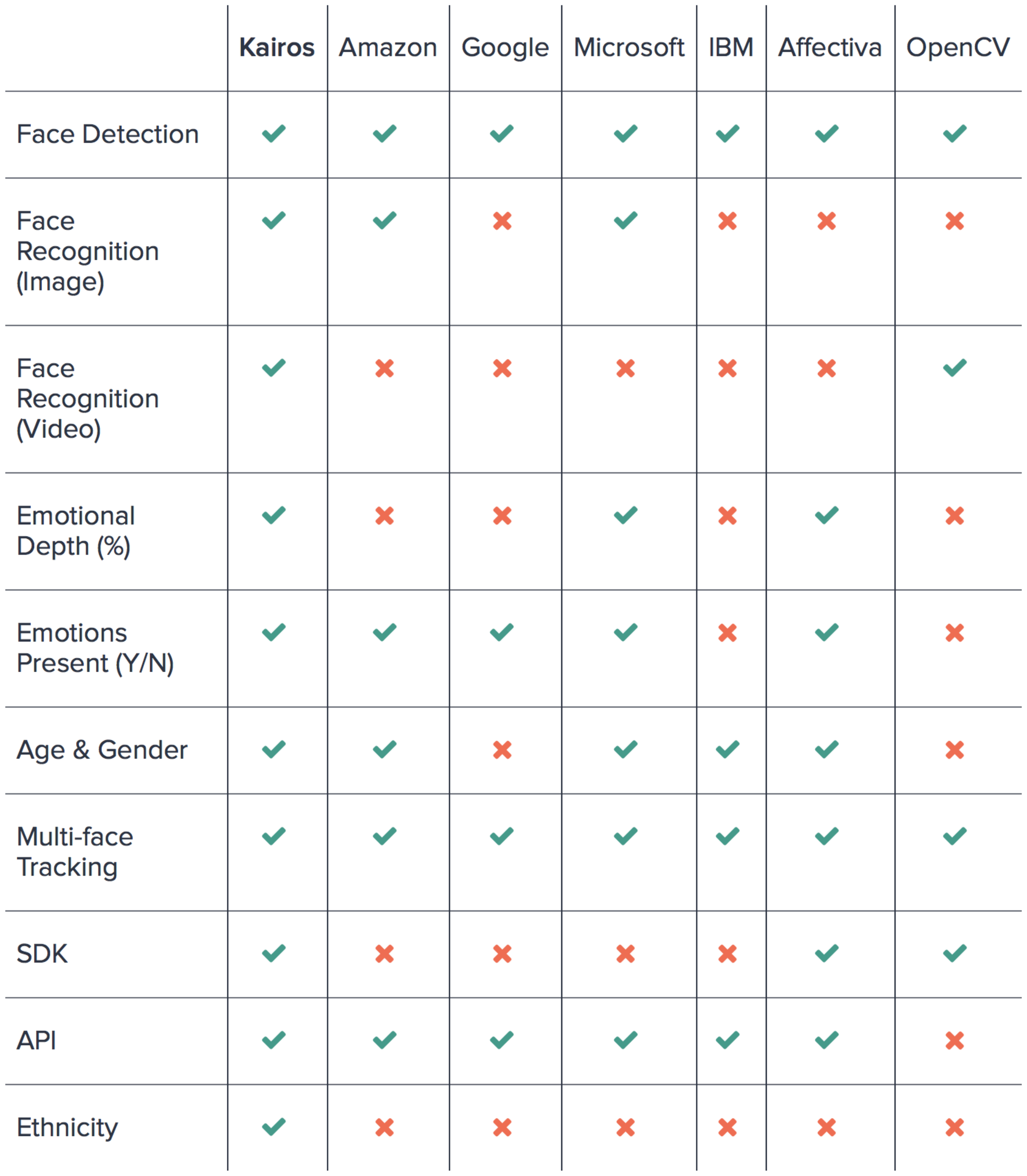
2.13.1 What is Kairos

Kairos is an artificial intelligence company specializing in face recognition. Through computer vision and machine learning, Kairos can recognize faces in videos, photos, and the real-world - making it easier than ever to transform the way your business interacts with people.And they have a nice team who works every day to improve their AI and bug fixing. Not only that they provide Online/Offline recognition as well. For this project I used the API they provides in order to integrate two factor Authentication using face as the authentication model.

The Karos API provides not only just facial recognition other than that it provides features like, Face Detection, Emotion Detection , Age Detection, Gender Detection, Attention Measurement, Facial Features, Sentiment Detection and Ethnicity Detection. According to them they have named this in a term called Human Analytics.

2.13.2 Advantages of using Kairos

Since I have tried out many biometrics solutions I feel this AI is the most accurate and ease to handle. They have also published an article comparing other available solutions in the market. The comparison they have done is as below.



	Kairos	Amazon	Google	Microsoft	IBM	Affectiva	OpenCV
Face Detection	✓	✓	✓	✓	✓	✓	✓
Face Recognition (Image)	✓	✓	✗	✓	✗	✗	✗
Face Recognition (Video)	✓	✗	✗	✗	✗	✗	✓
Emotional Depth (%)	✓	✗	✗	✓	✗	✓	✗
Emotions Present (Y/N)	✓	✓	✓	✓	✗	✓	✗
Age & Gender	✓	✓	✗	✓	✓	✓	✗
Multi-face Tracking	✓	✓	✓	✓	✓	✓	✓
SDK	✓	✗	✗	✗	✗	✓	✓
API	✓	✓	✓	✓	✓	✓	✗
Ethnicity	✓	✗	✗	✗	✗	✗	✗

Figure 6: Overview of Android architecture [20]

With this article we have a clear understanding on how powerful this kairos is and its benifits. And thats why this API is used as the two factor authentication in this development.

2.13.3 How the API works

You submit images or video into our API. We find all the faces and return detailed facial information back in the form of code objects (JSON). These results include feature points, emotion, age, gender, ethnicity, and a ton of other useful data about the human faces. This explains how Kairos API works. But in more they have labs that works parallely doing many test along with machine learning algorithems in making this solution the best available in the market. Plus just like other API or developments this API also encounts issues falts but their response team is really fast and they fix thing very quickly. Recently I found a issue in their cache servers and they fixed it immediately so its not just an API its an API that we can trust as they are involved on this every single day.

2.13.4 Proof of concept

In Regards to Face Authentication:
I have been using this API for almost a 5-6 months in my current working place for Attendance.
And we have over 15 employees and this worked perfectly for us. Also we have extended the development and used it for a client that has over 50+ employees and the results are positive.

By doing so I got to know what are the pro's and cons of this face authentication. In most case where a issue comes across internet is poor. And for this verification internet connectivity is a must.If you have a weak connectivity time taken to get the result will take more time.But in this development we provide a consistant WIFI router which has good internet connectivity so once the client is connected you don't need internet on the client app but through just wifi the data will be passed and from the local server will do the hand shake with Kairos API and get the job done.

In Regards to the Main topic Wifi Management:
Based on the experience and by performing minor analysis on WIFI password management the current practice is to connect to a network and keep connecting to it automatically or write the credencials and save somewhere safe. But this is not proper management. When ever the password is change the users will fall to a very difficulty situation. But after a test run and based on the results 90% accurate and the remaining 10% falls in to connecting issue. (Connect with the server)

2.14 Google Vison API

Designed and maintains by google and according to them Google Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API. And for this development live face detection, crop and blink event is captured by this API. Although this API has more important features but due to timing issues only few points were used in the development.

2.14.1 What is Google Vission API

"The Google Vision API provides a RESTful interface that quickly analyses image content. This interface hides the complexity of continuously evolving machine learning models and image processing algorithms."[24]

2.14.2 Available Features

- **LABEL_DETECTION:** executes Image Content Analysis on the entire image and provides relevant labels.
- **TEXT_DETECTION:** performs Optical Character Recognition (OCR) and provides the extracted text, if any.
- **FACE_DETECTION:** detects faces, provides facial key points, main orientation, emotional likelihood, and the like.
- **LANDMARK_DETECTION:** detects geographic landmarks.
- **LOGO_DETECTION:** detects company logos.
- **SAFE_SEARCH_DETECTION:** determines image safe search properties on the image

2.15 Overview of Software Components

2.15.1 Laravel 5.5

Laravel is one of the most powerful PHP framework designed security in mind that helps building complex web based solutions. If you are someone who looking forward to select PHP as the language Laravel is the framework you must learn.Also Laravel is a framework where the code is clean and modular and integration of many libraries are possible now. With the middlewere they provide it is easy to combine multiple modules and get thing done in a proper modular way.

Also Laravel provides just enough built-in features that you can focus on the more interesting parts of your app without having to worry too much about the boring basics.

For the development backend is designed with Laravel on top of SSL/TLS configured server also providing a web interfaces to show and feed data to client applications via the inbuilt API framework.

Also the inbuild security components is seconds to none.This feature really impressed me to use this framework without any hesitation. When storing password is is a common well know practice we used to follow and that is hashing. And why we use hashing instead of encryption is that in case of encryption if some got access to the key pair or the key all passwords are compromised.But hash is a one way function which means it is extreamely easy to generate the hash value of a text but extreamely hard for anyone to get the corresponing plaintext by knowing only the hash value.That is the basic theory behind it. And this also can be cracked with the help of dictionaries. In simple this means a library that contains commonly used plaintext along with their hash value making hackers to query and get the corresponding plaintext. A fix for this security issue is salting.

In salting what we do is that we take the original value append it with a unique or anyother value and that could be the username or the name itselft and then hashed which makes it really hard for even dictionaries to break these codes. In old days MD5 was a hit but with time MD5 is outdated and SHA has taken control. But as we know SHA need more text space that MD5. But fine what Laravel do is that it has its own algorithem that generate SHA hash and compress it to 60 charactors.And the beauty of this is that even though we have the same passcode for two different users the hash values are different. This is something really interesting where as in MD5 or general SHA algorithms same plaintext will have the same hash output.This is also another main consideration I looked in to in selecting Laravel as my Framework.

The above description explains my point of view on this framework. With the research's I have done several key points were found in regards to this framework. [14]

BUILDING AUTHENTICATION AND AUTHORIZATION SYSTEMS
Laravel makes implementing authentication very simple. Almost everything is configured out-of-the-box. Laravel also provides a simple way to organize authorization logic and control access to resources.

INTEGRATION WITH TOOLS FOR MAKING WEB APPLICATIONS FASTER
Laravel supports popular cache backends like Memcached and Redis out-of-the-box. By default, Laravel is configured to use the file cache driver, which stores cached objects in the file system. For larger applications, it is better to use an in-memory cache such as Memcached or APC. However, with Laravel it is even possible to configure multiple cache configurations.

FIXING THE MOST COMMON TECHNICAL VULNERABILITIES
Laravel helps to secure the web application by protecting it against the most serious security risks: SQL injection, cross-site request forgery, and cross-site scripting. Laravel itself is secure. We can tell you first hand that the codebase is fanatically guarded, and that the code has been vetted by several people

SEPARATION OF BUSINESS LOGIC CODE FROM PRESENTATION CODE
Laravel is an MVC framework, so separation is already done. See the figure: the full MVC request cycle in a Laravel 5 application.

2.15.1.1 Eloquence

Laravel uses its own ORM, called Eloquent. simple ActiveRecord implementation for working with your database. Each database table has a corresponding "Model" which is used to interact with that table. Models allow you to query for data in your tables, as well as insert new records into the table.

2.15.1.2 Why selected Laravel over other frameworks

With the introduction I have provided based on my research showed me some positive results and for PHP development I'm new but I never felt that way in dealing with Laravel so I belive I made a correct choise.

2.15.2 JSON

JavaScript Object Notation (JSON) is a light-weight data-interchange format.

The advantages of JSON are:

- Easy to read and write.
- Easy for machines to parse and generate.
- Completely language independent but uses conventions that are familiar to programmers and can be implemented in languages like C, C++, C#, Java, Perl, Python, JavaScript, etc.
- Unlike XML, JSON is not verbose, faster and consumes lesser resources. It is therefore suitable in low-resource environments.

Data communication with Android to Backend backend to Android is all based on Json. Result set is mapped to Json (Object & Arrays) from databse tables and shared through web services.

2.15.2.1 JSONObject

The JSONObject class stores the unordered set of key/value pairs. It has the following main methods:

- **put():** method to add or replace values in an object.
- **get():** method to retrieve values from an object. This get() method is of generic type and returns value of type object which can be cast or queried for type.

The JSONObject class also has typed get methods that do the type checking and type correction. These methods are used to return a specific data type; getInt(), getLong(), getBoolean() , getJSONObject(), getJSONArray() are some examples of typed get methods.

2.15.2.2 JSONArray

The JSONArray is a class that stores an ordered sequence of values. Each element in array is of type JSONObject. It has get methods for accessing the values by index and put methods for adding or replacing values. A JSON text can be converted to a Java object in the constructor and from Java object to JSON text using toString() method. Elements in the array are separated by comma (,). The get and put methods work similar to the methods in JSONObject, but take an index value instead of a key as first parameter.

2.16 Android SDK

Android Software Development Kit (SDK) provides the necessary API libraries and tools for developing, debugging and testing of Android Applications. The SDK consists of various modular packages that can be downloaded separately using its Android SDK. Also they provide set of inbuilt API that can be used to design and develop android applications.Every single year with Google release a new android version so to support that an SDK is introduced.So with the SDK all the needed functionalities are bundled together and integrated in to a one single component to help the developers build quality android applications.

2.17 Retrofit

Retrofit is known as a A type-safe REST client for Android and Java which allows the developer to build powerfull applications which interact with network services with ease consuming less memory. If we tried the native way of doing this this could take more time to integrate plus the maintainability of the code is a mess and the number of lines of code I need to write is also not less. But this can be solved with this library.

"Retrofit is a type-safe REST adapter that makes easy common networking tasks. Retrofit turns your REST API into a Java interface and uses annotations to describe the HTTP request. Retrofit takes care of URL manipulation, requesting, loading, caching, threading, synchronization... It allows sync and async calls.Retrofit is among the most used and most famous REST clients in Android." [15]

2.18 Green DAO

GreenDAO is an ORM (Object Relational Mapping) designed by greenrobot.org on top of SQLite allows the developer the designed and build great apps fast and cleaner. And when the performance is considered there are times where GreenDAO passes SQLite benchmark and proves the world that this is currently the best ORM designed that can match with SQLite but seconds to Realm in some areas. Realm is currently the fastest of all as it is designed with c language.

2.18.1 ORM Performance

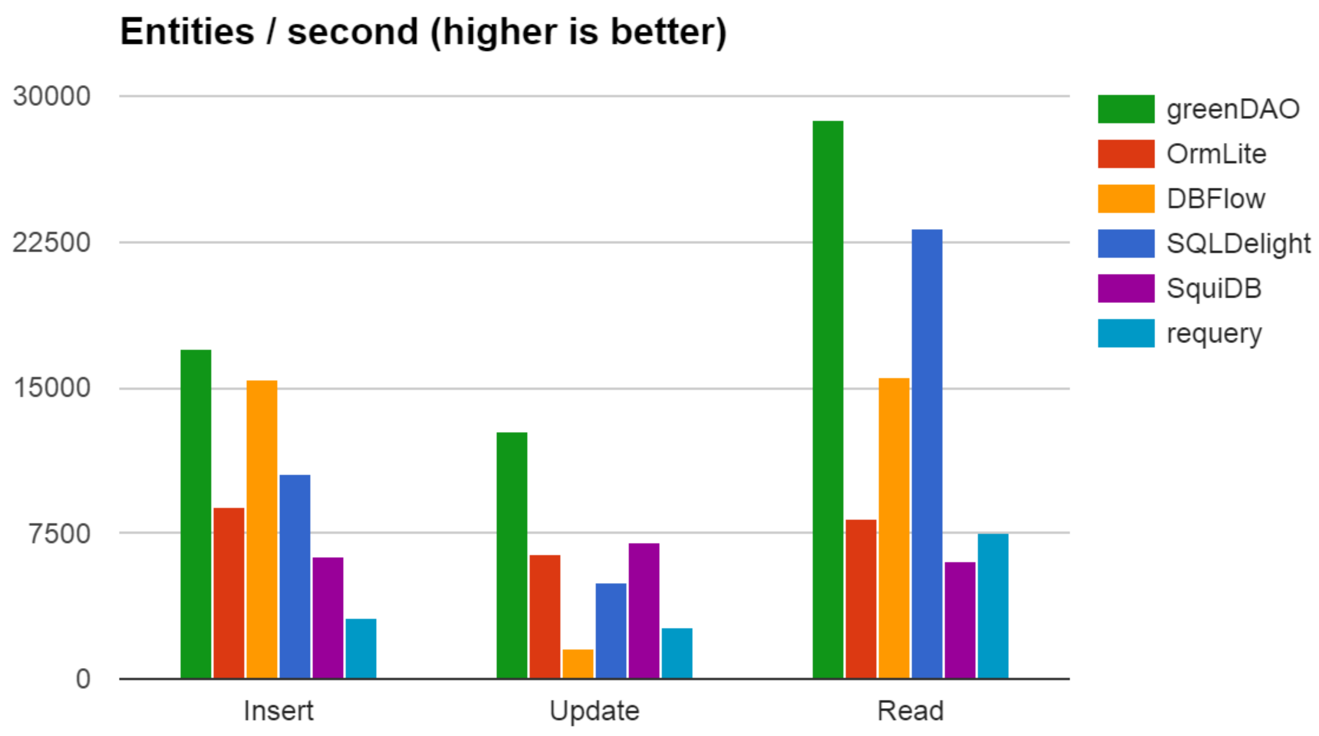


Figure 7: Android ORM Performance [21]

Based on the above figure As we can see greenDAO comes in first in for all operations while there is no distinct second place. And with the experience I had on this GreenDAO is selected for the development.

2.19 Android Schedular

JobScheduler is the Android framework API for scheduling tasks or work. It first became available in Android 5.0 (API level 21), and remains under active development.The best factor about this API is that we can trigger events by applying filters. Like call and run if the device is connected to a powersource/Network etc.

2.20 Java

"Java is a computer programming language. It enables programmers to write computer instructions using English-based commands instead of having to write in numeric codes. Its known as a high-level language because it can be read and written easily by humans. Like English, Java has a set of rules that determine how the instructions are written. These rules are known as its syntax. Once a program has been written, the high-level instructions are translated into numeric codes that computers can understand and execute."[25]

2.20.1 History

"In the early nineties, Java was created by a team led by James Gosling for Sun Microsystems. It was originally designed for use on digital mobile devices, such as cellphones. However, when Java 1.0 was released to the public in 1996, its main focus had shifted to use on the internet. It provided interactivity with users by giving developers a way to produce animated webpages. Over the years, it evolved as a successful language for use both on and off the internet."[25]

2.20.2 Why Java

Java is one of the best programming language ever built and on top of the language is famous due to some key main factors.

- **Ease of Use:** The fundamentals of Java came from a programming language called C++ as c++ is place where people start shifting developments from standard procedural way to object oriented way. Although C++ is a powerful language, it is complex in its syntax and inadequate for some of Java's requirements. Java built on and improved the ideas of C++ to provide a programming language that was powerful and simple to use.
- **Reliability:** Java needed to reduce the likelihood of fatal errors from programmer mistakes. With this in mind, object-oriented programming was introduced. When data and its manipulation were packaged together in one place, Javas was robust.
- **Security:** As Java was originally targeting mobile devices that would be exchanging data over networks, it was built to include a high level of security. Java is probably the most secure programming language to date. When it comes to banking a first priority is given to java for its proven security capabilities.
- **Platform Independence:** Programs need to work regardless of the machines it is being executed on. Java was written to be a portable language that doesn't care about the operating system or the hardware of the computer or device it is running on.

2.20.3 JDK

Java Development Kit is the tool kit that is used to design and develop java based application. In contains set of predefined libraries that enables users to develop secure applications. Also it contains the runtime environment which allows the PC to run java based application on something called a sandbox in making it platform independence.

In simple this kit contains all that is required to build debug softwares that were built using java platform. If this kit is installed any applications written on java can be executed as well but this tool kit is not seen in production level but is a must for development.

In regards to platform independance this can be installed across platforms like Linux, Windows & Mac operating systems.

2.20.4 JRE

JRE stands for Java Runtime Environment. By default if JDK is installed this is included automatically as this enables users in running java based applications. And why this is more usefull is that if only this module is installed the user will not be able to do any developing but java based applications can run with out any issue. So we use JRE for client machines where no development is needed.

This toolket is more towards running java application and consume less space and memory which drives the developers to run their solution in a multiplatform environment using a common runtime environment where the same code will work everywhere concept.

2.20.5 Approach

First as the start up a set of questioners will be shared with friends and others to clearly understand the current problem along with their suggessions to minimize those issues. Once these data is collected the next step was to create a proper scope that could be covered with in the given timeline.

As per the development the Wifi Manager application client was developed using Java Software Development Kit 8 on Android OS. The integrated development environment used for the writing Java classes is Android Studio.

Development frameworks like Retrofit was be used to communicate with webservices and third party API Kairos will be used to integrate face authentication where as the face detection and blink detection will be handled by a custom algorithem plus google Vision API.

Chapter 3

3 Design

3.0.1 Application Architecture

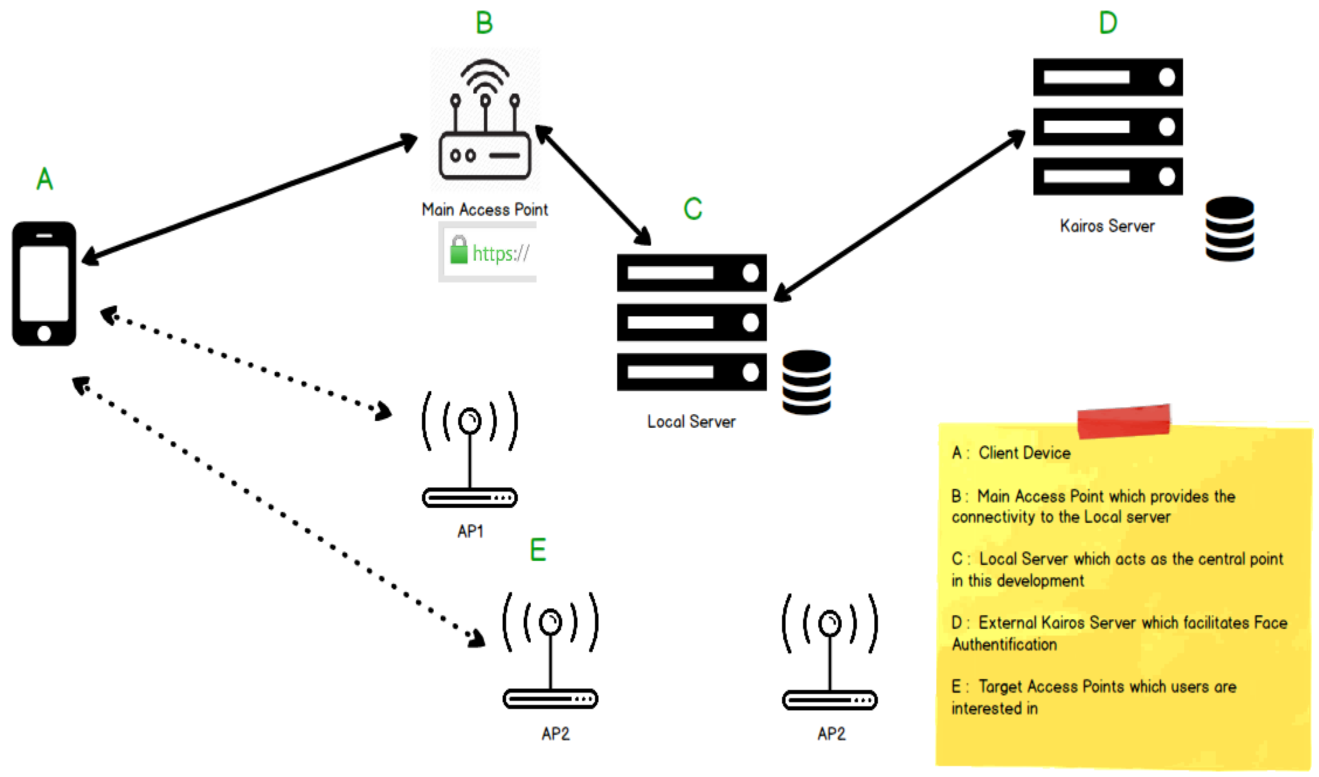

B
D
A
Main Access Point
https://
C
Kairos Server
Local Server
AP1
E
AP2
AP2
A : Client Device
B : Main Access Point which provides the connectivity to the Local server
C : Local Server which acts as the central point in this development
D : External Kairos Server which facilitates Face Authentification
E : Target Access Points which users are interested in

Figure 8: Application Architecture

This solution is a combination of 5 key components as mensioned in the above diagram. First component is the Client Application which comunicates with other modules get the work done which is indeed to connect with another WIFI access point. In order to do so the device must connect with a intermediate Access point which will provide a direct link with the server and the client. This is done by the intermediate Access point which is configured to a local server and the communication happens in a secure medium where it will streangthen the development from attack possibilities.

The Solution is designed to capture users credentials along with the face ID which will be processed at the local server where as the face ID is computed and verified at the Kairos server. Once the authentication is finalized all access point info is transfered in to the client devices in encrypted format which uses symmentric encryption mechanism which makes it hard to decrypt in case of a data theft.

Frontend client application is designed with Android mobile development framework and the Local server is designed with PHP Laravel framework.

3.0.2 Application Flow Chart

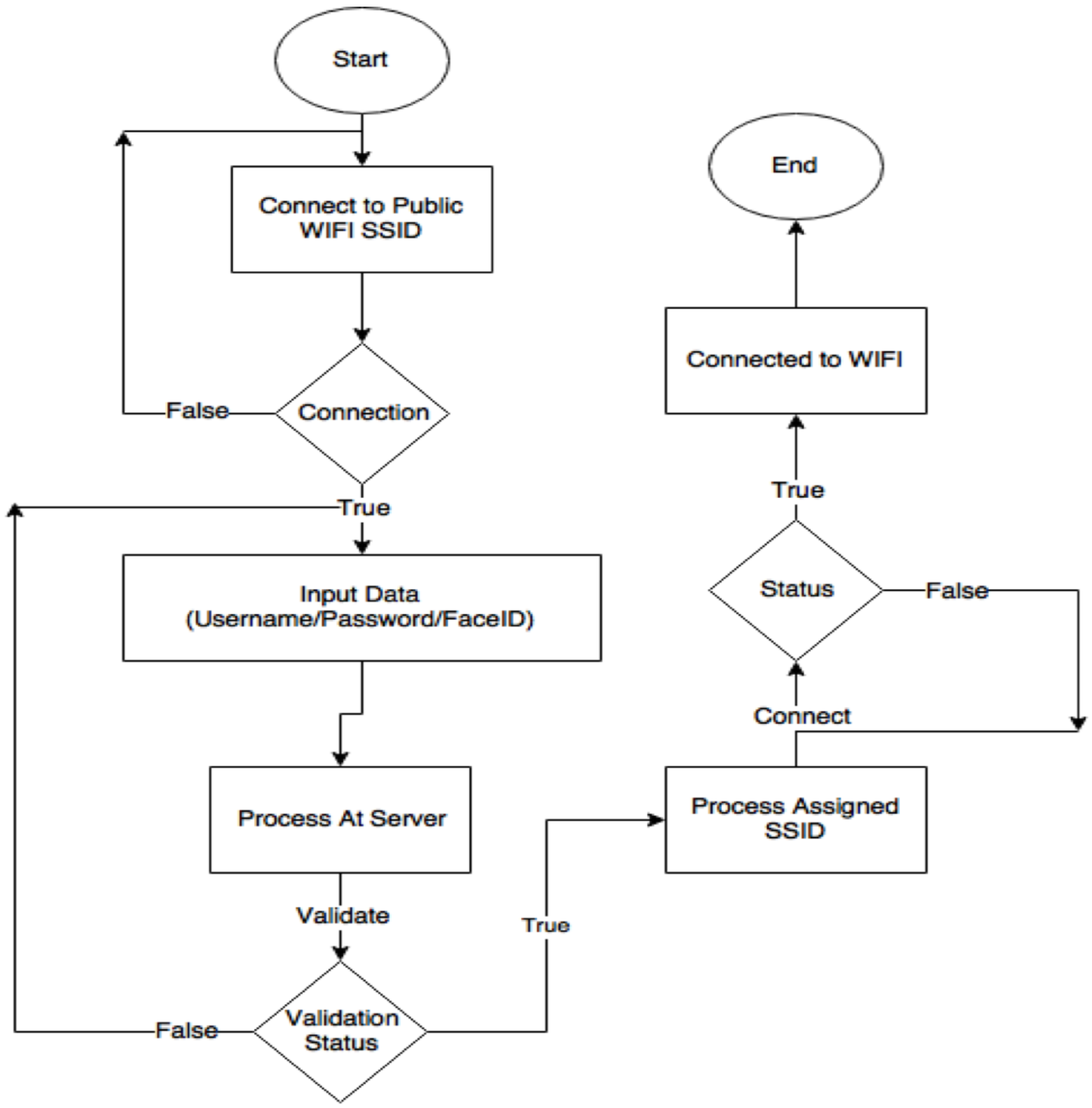


Figure 9: Basic Flow Chart

From the chart above, the basic process is briefly explained from the place where the user tries to open the app to conneting to a asigned wifi network. First user will be connected to a public wifi network where the server is conected. Once the connection is made user needs to input his/her credentials and verify his/her identity. Once identity is verified second step is also checked if the user has enabled face authentication. Once the user is fully verified its asigned wifi networks will be downloaded in to the application. From then the user is required to select a wifi accesspoint and use the network until the user disconnects.

Chapter 4

4 Implementation of wifi Management Application

4.1 About the application

Wifi Management Solution is a development targeted to provide WIFI access to android phones via android app with advance user tracking two factor authentication designed in security in mind In short how this solution works is as below. A user can only be created from the server only by a administrator. Only the Administrator has the power to add Users,WIFI access points asign users to wifi SSID's etc. And all the events are logged in a activity table for tracking.
For security purpose wifi password is encrypted and saved in the mysql database with a symetric key of 128bit key. Also the password of the user is again hashed and stored in the database.

User will download the application from Google play and on Login event first it will connect to the nearest WIFI Accesspoint which is also connected to the server. And input from the user is taken and passed to the server via https connection and authenticated. If a user is found check is performed to see if the two factor authentication is enabled of not.If enabled face is captured and passed to the server and verified and direct to the respective Dashboard.

If a ordinary user other than Admin is logged they will be directed to user Dashboard where assigned WIFI networks are visible. On click event the phone will be connected to the WIFI network and the app will be closed. After 1hr a revalidation is required to maintain the connectivity.

If the Admin is logged in to the system the admin can enroll a face of a existing user from the app. View Activities from the app.

4.1.1 Client

4.1.1.1 Introduction

The Client android application is the main user interface that the user interacts with. Which simply allows the user to use their credentials to gain access and get accessed to asigned WIFI networks.

What is special about this client app is that it requires the app to be running on Device Administrator mode. Which means the application cannot be uninstalled or forced stoped after login in to a WIFI network. This will come in handy if the user is trying to fool the system and also wants to overcome the 1 hour timeout to refresh the connetivity.

In this case if the user tries to deactivate the app the application is going to start automatically and will clear from connected WIFI access points.

Once the user accepts the Admin mode and decides to continue the android device will get connected to the common WIFI accesspoint automatically with out any user interation.Once connected to the network now the application can interact with backend REST api over https connection where the data is encrypted so that this would avoid evesdrop attacks. Now the user has to feed his/her password and username as we primarily rely on the user credentials although we know that the credentials can be passed to someone else. But to clear that doubts for selected people we can add two factor verification using Face recognision.

The reason behind not making this mandetory is that in an organization there are upper management people who likes to get things done easy and quikly for them they can disable and use on the user credentials.

So with the user credentials the user is authenticated and as the next step if the two factor auth is enabled the user is required to scan his face using the application we have designed. The beuty in this development is that simply facing tha camera won't do a thing to avoid user playing with the system by using photographs a blink is required to capture the event.On capture the the data is passed to the backend and get if the resultset. This is again compared with the results we received with user credentials if both match all the available WIFI accesspoints are saved and stored for the next process. Here the WIFI ssid is saved in plain text but the password is saved in an encrypted format using 128bit symmetric encryption.

Once the user is logs in to the User dashboard the very first thing the user sees is that the available SSID for the user to connect. Simply what the user needs to perform is select the desired wifi network and wait till it gets connected. On connected event a time will be placed for an hour of time to re authentication of the same process and the application will be closed.

On each events where the user is interacting with the Server a log is recorded to support Accountability so that the user who used the system is accountable for any actions he performs under its login.

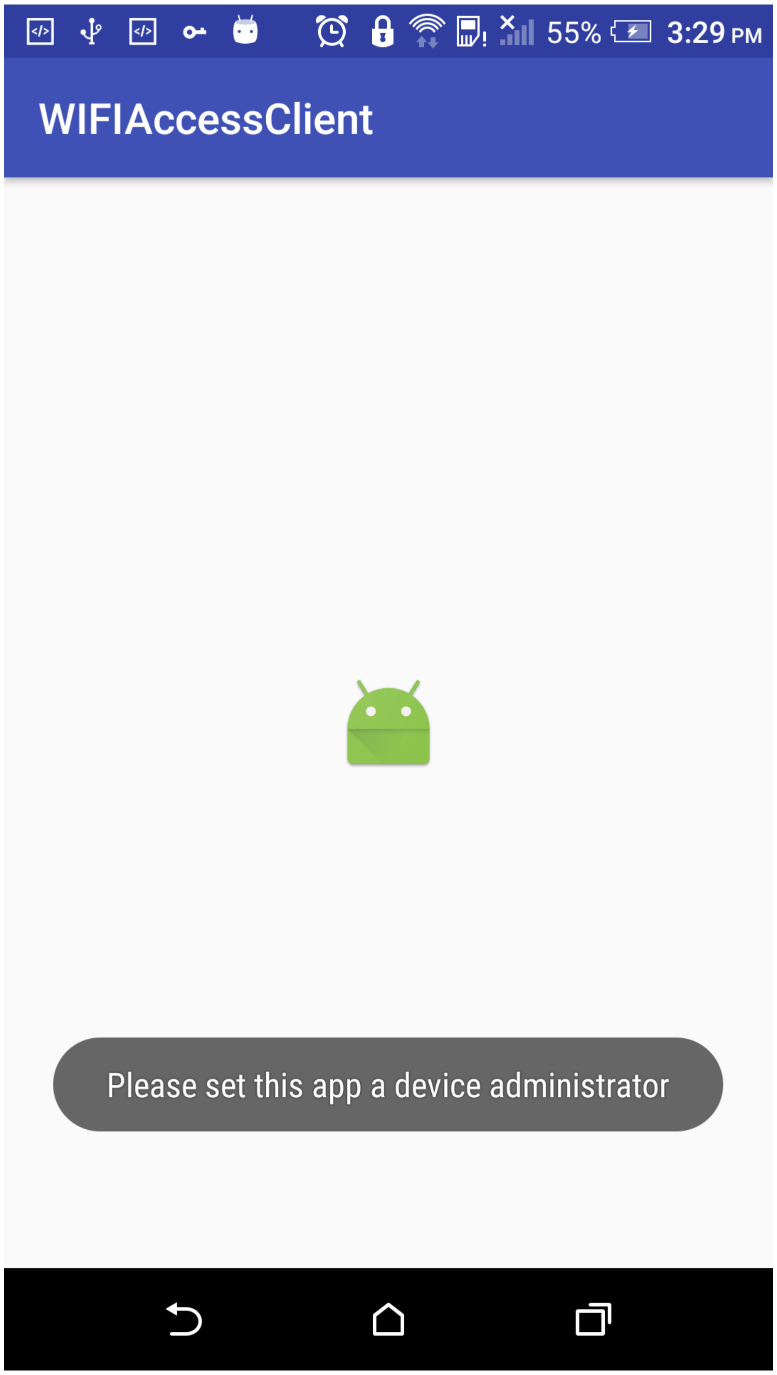
4.1.1.2 Interface wise Description

Splash Screen:

This is the first screen that comes when the application starts. On this screen it is first verify the app admin status. If in case the administrator more is not activated a popup showing in order to activate the application to administrator mode.
Once activated only the next step is available which is to maintain connection with the default Wifi SSID in order to connect with the server.Once connected and the administrator mode check is done next window comes to the display.

Most of the security checks and validations is done with this screen like checking if the app is running as the administrator or if the device is connected to the nearest WIFI ssid if the permissions required to run this app is met etc.So although this interface does look simple the process behind is not.

The main reason behind making the application as the administrator is that if not the user might try play with the system like once the user is already connected to the system in order to avoid 1 hour re authentication the user will try to force close the application so that the receiver that has been programmed to run after 1 hour would stop allowing the user to gain full access to the WIFI for a long period of time.

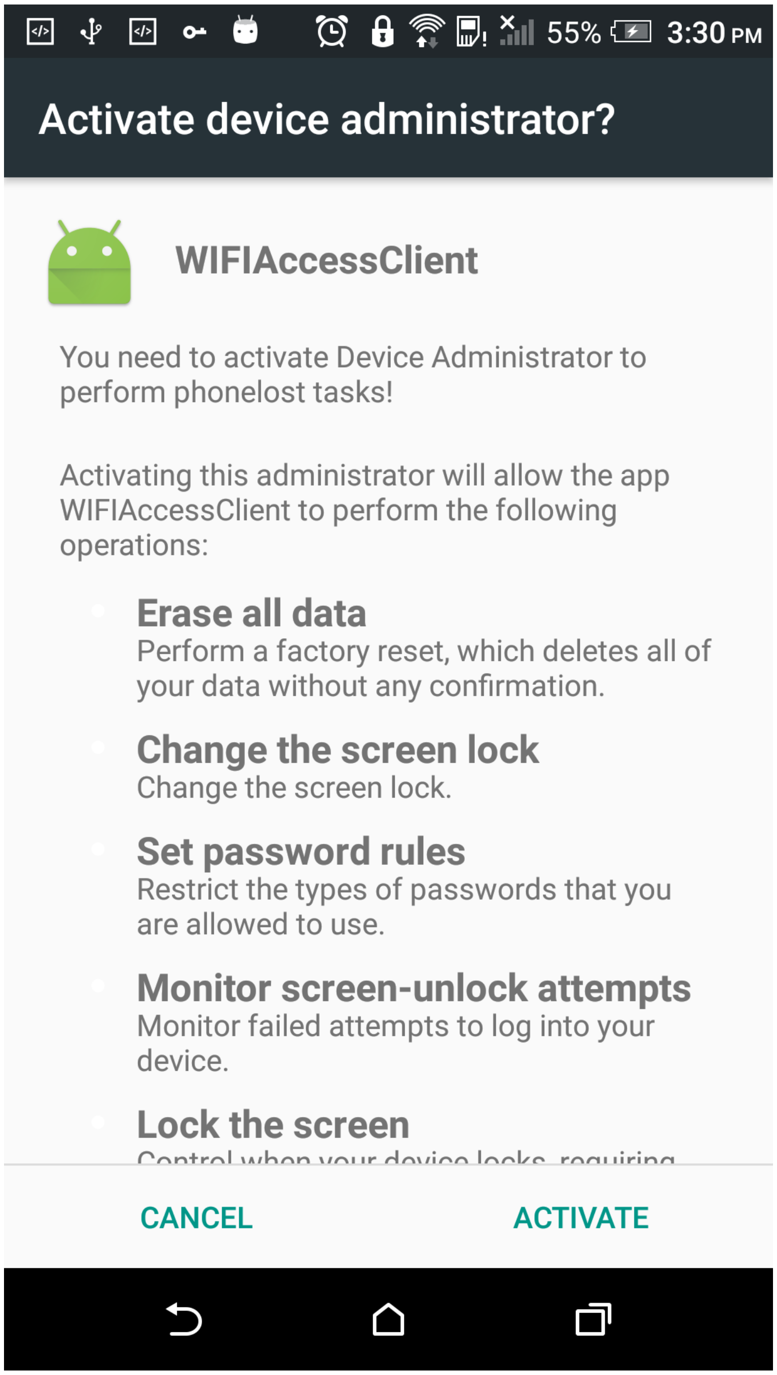


Please set this app a device administrator

Figure 10: Splash Screen

The screen aboves is shown if the current status of the application is not admin mode. So after words the user is directed to the below screen.

As it is shown if the user has not set the admin mode the above message will be shown on the screen and then the user will be directed to another interface where the user has to enable the administrator mode to proceed.



You need to activate Device Administrator to perform phonelost tasks!

Activating this administrator will allow the app WIFIAccessClient to perform the following operations:

- **Erase all data**
  Perform a factory reset, which deletes all of your data without any confirmation.
- **Change the screen lock**
  Change the screen lock.
- **Set password rules**
  Restrict the types of passwords that you are allowed to use.
- **Monitor screen-unlock attempts**
  Monitor failed attempts to log into your device.
- **Lock the screen**

Figure 11: Admin Activation

In order to use the application the user first has to enable the admin mode by doing so the user is allowed to use the functionalities provided by the application.

The reason behind this process is to make this development bullet proof from cracking attempts. Basically this would not allow the user to clear app cash or data and the most dangerous thing can be done to the application, uninstallation.

Login Screen:

In this screen the user is required the feed its username and password and hit submit. Then a REST webservice call is triggered with the data attached and waits for a response. Here there can be so many scenarios like the username & password is incorrect or the user does not exists etc. So from the logic the success and error is properly identified and a meaning full message is shown. In case the credentials are correct the user gets a responce and requirement to use two factor authentication is checked. If the second layer of authentication is required the user focus is change to the next screen or else Dashboard is shown.

This is the most important screen where if the security is again considered a Login Screen. This will still act as a simple page that would allow the user to access the application if the correct user credentials are used. And for this interface basic requirements of a login screen is matched. The password field is not showing the used password and will not allow if both fields are empty and proper validations are put in place.And un like other apps a forgot password module is not added here so incase the user can no longer remember the passcode he/she can contact the administrator and reset the password.

By this time the user is already connected to the predefined WIFI ssid but if incase the user leaves the screen other than proceeding further the connection will be removed and the user will no longer have the connectivity to communicate with the server or Internet. And the users might think that once connected to this default network we dont need any other network.But the point is correct because in order to make face recognition a web request is required meaning the connectivity is again required. So it is a must to make sure the user the people who tries to connect to this network is actially using the purpose. But to make things easy we could do two things. One make this a Public network and only allow the server to get passthoguh a firewall and access the internet for rest the wifi connectivity will be available but the internet connectivity is not.Second once the user leaves the screen the connectivity should be gone and the password used for this WIFI SSID should be changed more frequently so that even if the wifi is already connected once the password is changed the user is required to change its password to proceed.

In regards to the authentication there are two things that the admin can select when adding or modifying a user, if the face authentication is required or its not. Simply we have given this option is for the people who thinks the additional layer of security is not required and they always prefer the passcodes to access things quickly without wasting more time on the aditional step.

If the face authentication is not enabled based on the users credentials status available assigned WIFI neworks are downloaded and stored for further reference. If not the user is taken to another interface so that the second layer of authentication can be performed.

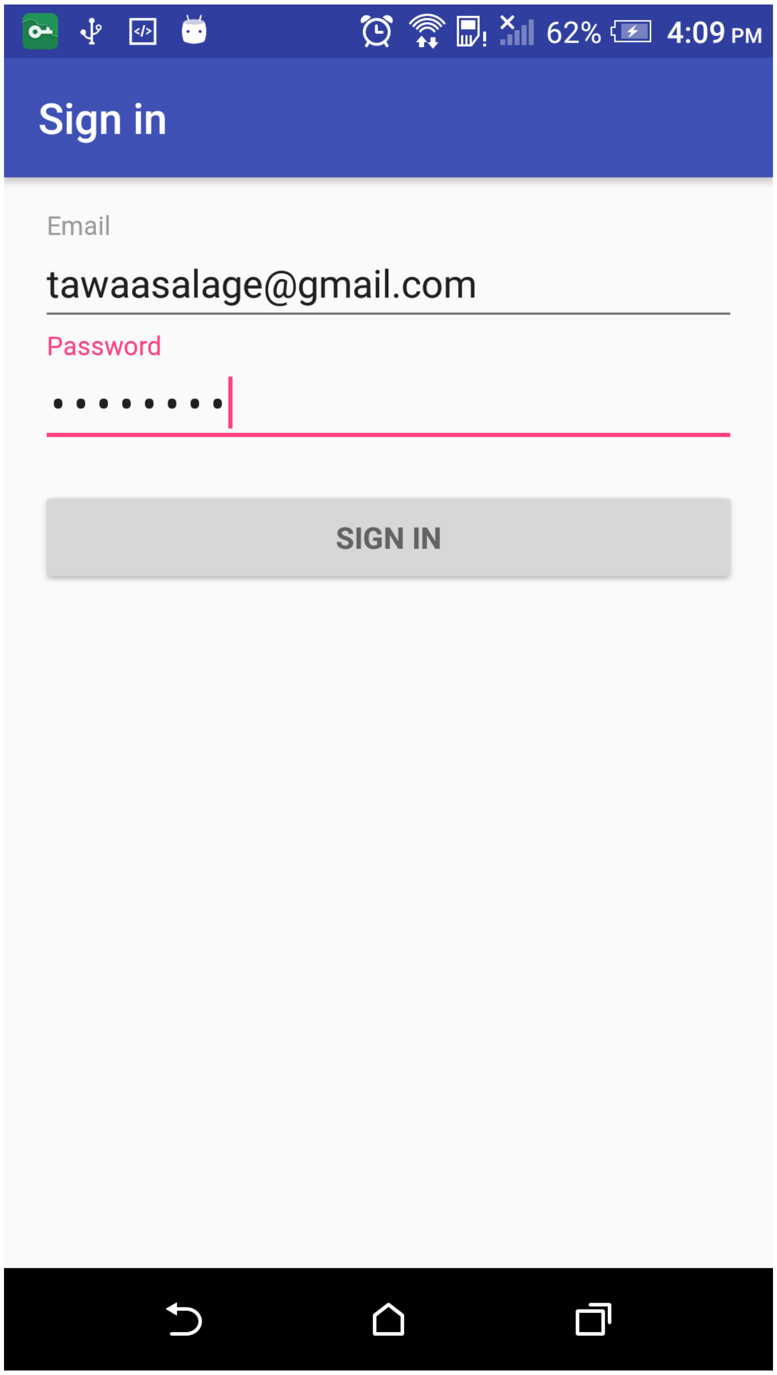


Figure 12: Login Screen

User has to verify his access to the application by providing valid informations in this screen. Username and passwords must feed in to the system. If the user has enabled the two factor verification user is then directed to the Face recognition module for second layer of verification.Else the user is directed to Dashboard based on his/her login priviledges. In our system the users we have divided in to two main categories. Admin & User.

Face Verification:

In this screen a Live camera is shown and a rectange forming if a dace is detected. Once a proper blink is detected a image is captured from the front camera and passed in to the server.
Here also the server provides set of responses. If it is valid the associate user info is passed back. So then the user is matched and if the user is who he claim to be with user credentials, respective wifi accounts are downloaded and saved in the local database.By doing so the password associates to wifi SSID is encrypted using 128bit Symmetric key to make things secure.

This screen is defined in order to capture and verify the user based on the characters of his/her image (Face). How this works is that when the user place his/her face in to the capera a unique face id is formed along with a rectangle with a unique color. This face id and color represents a one unique person in front of the camera. If the focus is lost for the same person another face id is formed. This allows us to capture one single photo of the same person untill the face id get changed. Simply putting the face is not enoght to make this work. A proper blink is required so that we can assume that the user infront of this camera is a person not a image. How we have programmed this blick is that the user needs to close left and right eyes for 60% and open it up to 90% so that we know the user has blinked. Once this is detected in that milisecond a image is captured. So again this image is parced to a function and the quality is reduced and the face is selected and cropped from the image for better transmission. There are two sides for this if the quality of the imageis considered, the better the quality more acurate it becomes. But the disadvange in this is that this will take more time in getting the responce as we have more bytes to transfer. However in Kairos API what they suggest is that the image should be around 28k for better transmission. So with this we have achieved the requirement from the vendor.

Once a proper face is detected and trasmitted to the server the server will communicate with Kairos API and them come with a result. Once the responce is reached to the android app the function is written in order detect if the user is allowed to proceed or not.Consider a scenario where a user input credencials for the user Nimal and on face recongition by placing Kamals face should not allow anybody to access. This control is placed in. If both conditions are matched the user is allowed to direct to its prefered Dashboard. But if the user is not asigned to any WIFI network still the user not allowed to navigate to the dashboard as he/she has nothing to do with the app then.

When security is considered a mi direct using a picture is blocked however still a video can be used to fool around with the system. Still thill can be avoided with an aditional API but I will not be using that as it consumes more time to process. Simply this chandles by studing what is around the picture.

At each time the user tried to ACCESS A WEB API a log is placed so that in case we need to go deep in case of an emergency it is helpful.

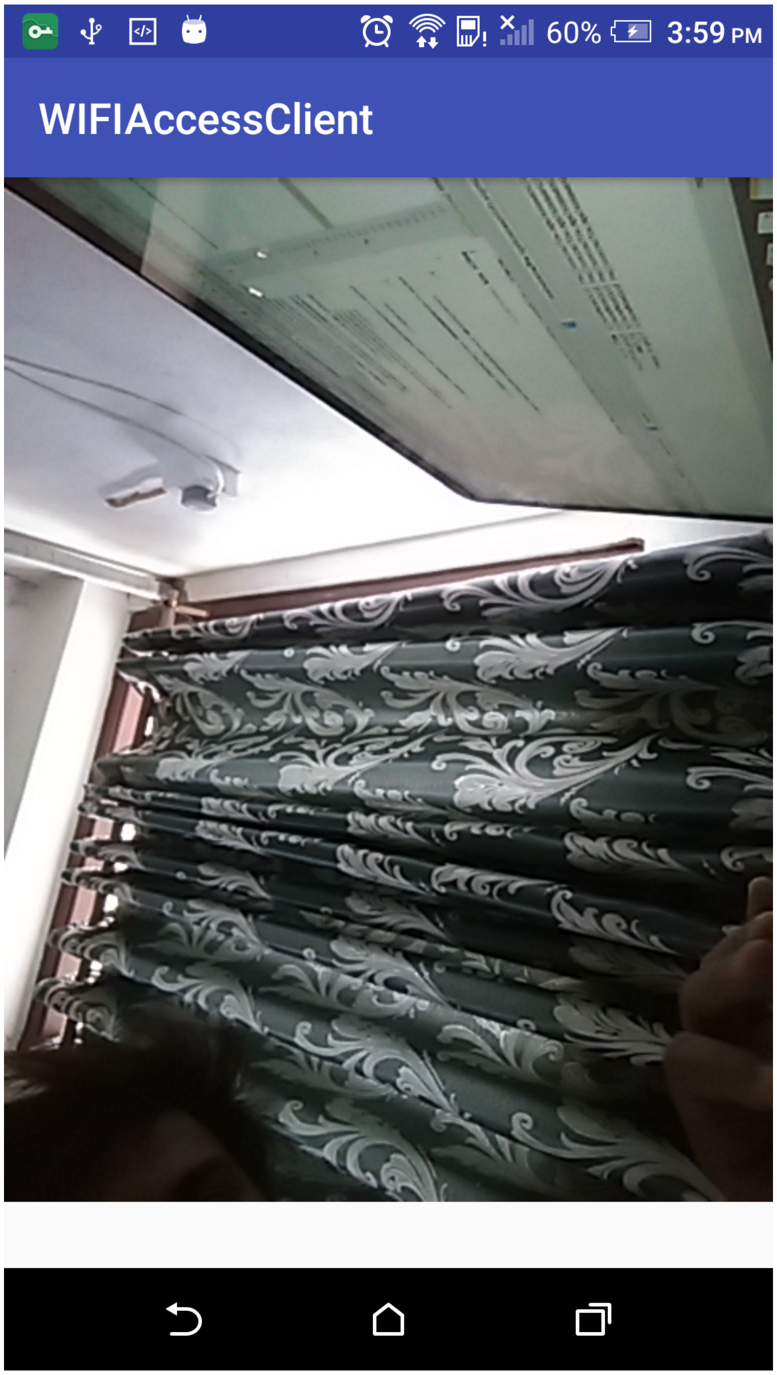


Figure 13: Face Recognition

The above screen determins whether the user is blinking or not and based on his/her trigger user is validated upon his/her provided credentials to verify the user.

Once a face is detected first the face will be recognized and a box will apear around the face with a unique color. And this color will be change once a new face is detected of if the camera lose focus with the face. Once detected a lenght from the device to face is detected and if this is closer the development will wait for a blink command. Here the blink command is given to avoid spoofing with images. Once detected the image is croped and feed to a external function for analysis.

Dashboard:

On a successful log in the asigned WIFI accounts are shown in a list where the user has the ability to select a desired wifi network. Althogu the user has no idea on the password what so ever the code is written in a way that the password is decrypted and connects to the network automatically and a dialog is shown to notify the user about the interval. Then the application is going to close and runs on background. In the event where the user was not connected to the network app will not close untill the user is connected to the network.

This Dashboard is designed for one single reason. To connect to his/her prefered networks. Also the user has the ability to change the passcode if required but changing the username or the email address is not allowed. But this is additional task which still requires the previous password so that we can confirm the person who tries to perform this task is acturely who he is.

If the user decided to connect to a Defined network simple clicking Connect button which is on the side of the network SSID is enough. By doing so the app tries to connect to the wifi network. If connected the user is notified with a Dialog saying after 1 hour re authentication is required and will close the app. If app could not connect to the network after 5 complete attempts a dialog with a Error note is displayed when that occurs the possibilities are the WIFI SSID you are trying to connect is off on not nearby.If you have any further questions forward a support request to the administrator.Once the user decides to connect and it gets connected the app will be closed and you dont required to re open the app untill it does it for you afer 1 hour.However if you re open the app before 1 hour whole process is again continues.

Next main functionality available is to change a password. We know that it is a practive we change our password and never to keep the default password. So changing of the password is something we look forward.But in doing so we cannot thest let anyothe do it.First we need to get the password of the current user then the new password and its confirmation is also taken into consideration. Once the basic validations are done the request is trasfered to the API where it will do the backend validation and change the password if the validation criterias are met.Once this is done if the password is changed the user will be getting a responce based on the status from the server so from next login onwards the new password is used. And this ends the two majo things that can be done with the Dashboard currently.

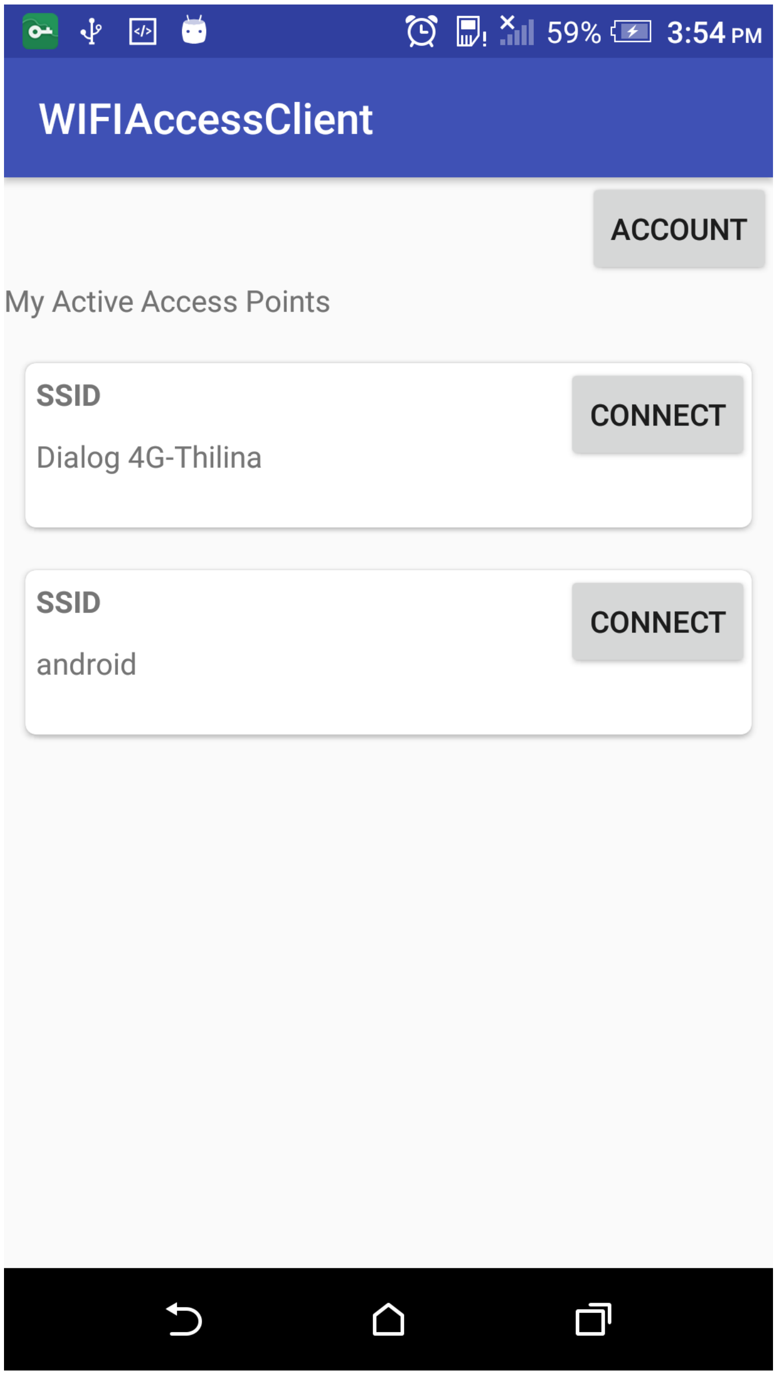
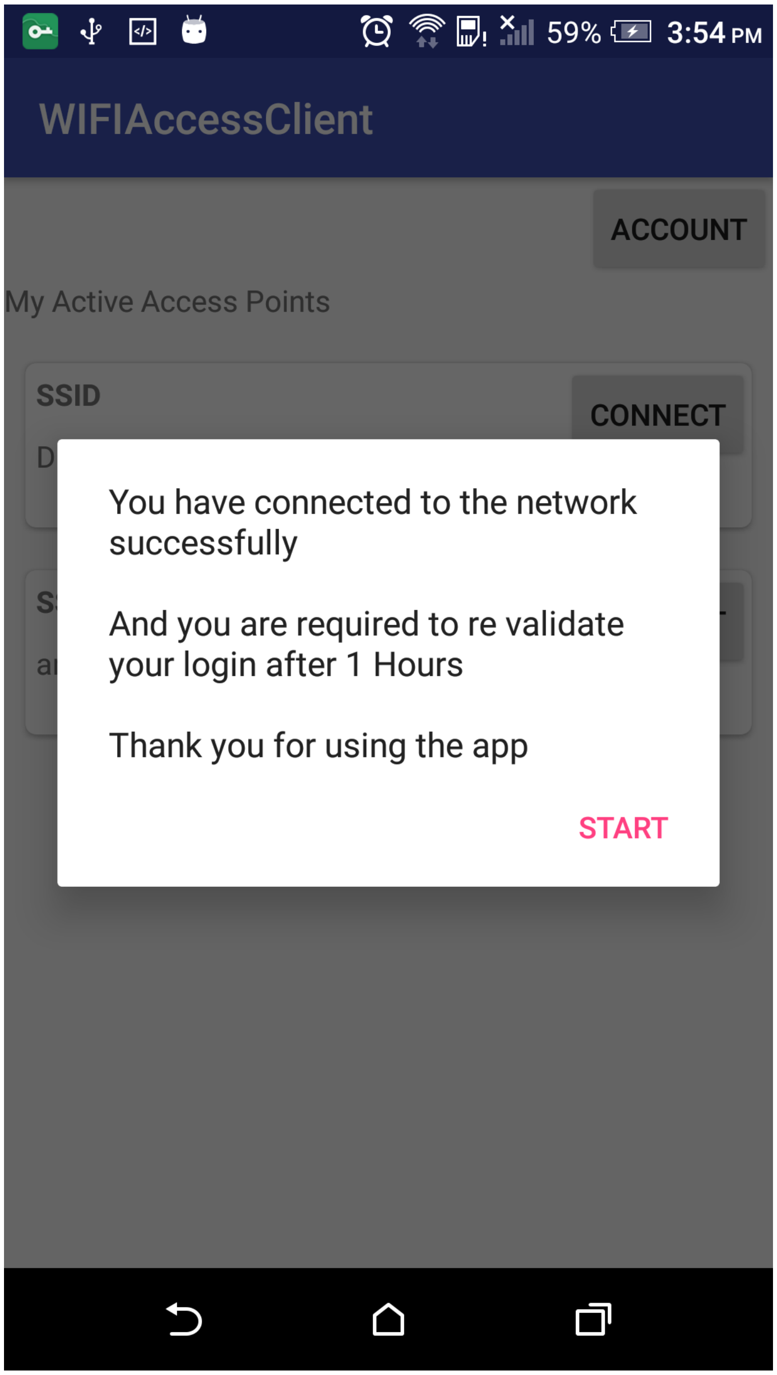


Figure 14: Dashboard

Once the user is logged in to the system this is what appears in his/her account dashboard, asigned WIFI networks. In order to get connected the user must select a valid network and click on connect button.



You have connected to the network successfully

And you are required to re validate your login after 1 Hours

Thank you for using the app

Figure 15: Connected

Once the user is connected to the network correctly based on his/her timeout selection user is notified. This timeout acts as the time you are allowed to use this network in hours. You could define user's timeout from backend. But only the admin has such priviledges to modify.

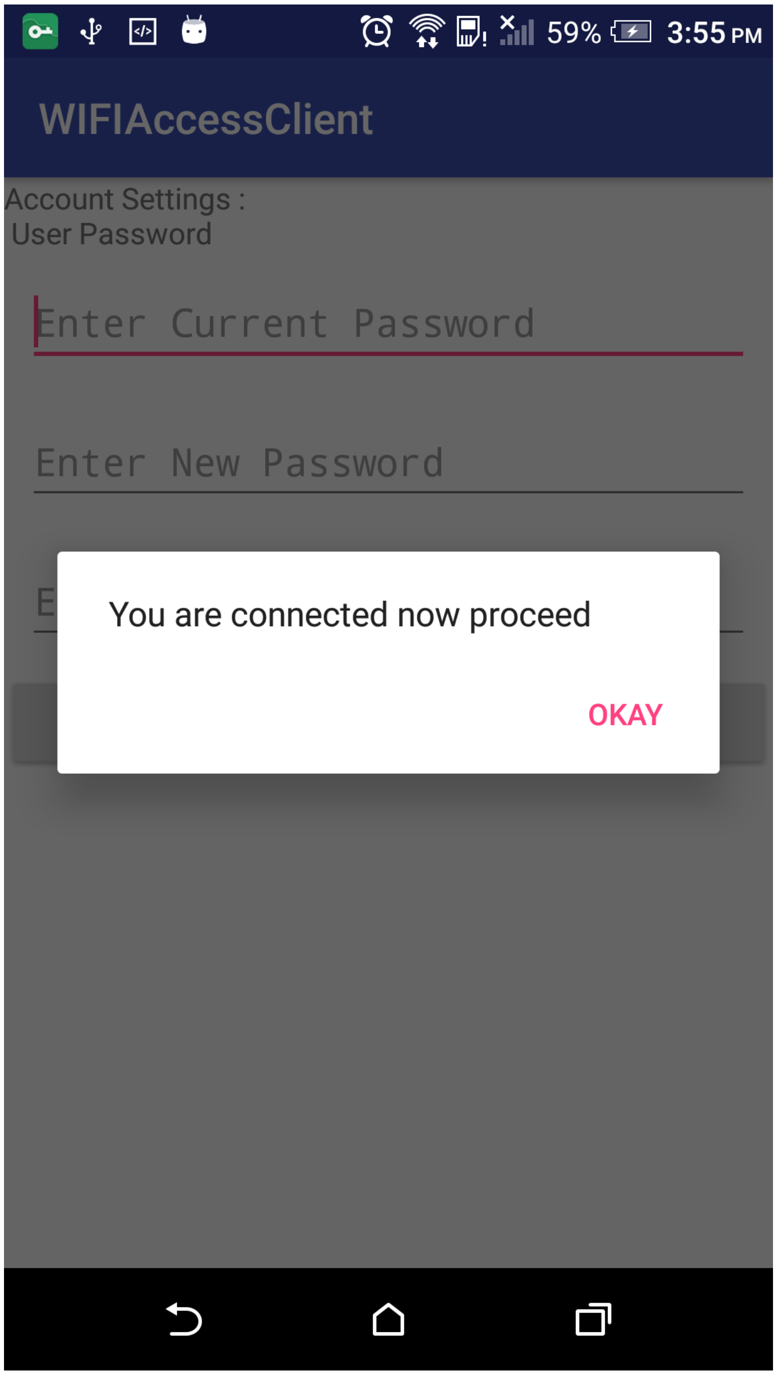
Account:

As we know a basic need of a user in any form of application is to put a secure passcode for his/her account. Without this feature any application is a joke if they keep the same password forever unless the password is unique and shared with only the user itself. But this is practically not hapening as a result we look for a feature like this in any application.In providing that the app itself has a functionality to change its default password. But in order to do so the user has to provide its current password along with the new passoword. And as a validation password is set to a minimum of 8 charactors and the new passcode is again verified with another input field that avoid simple human errors. But in order for this to work the app must communicate with the server.

But when the user logs in to the Dashboard the default wifi ssid is removed from the device. When the user first clicks the Account button to navigate to account screen in background the device is again get it connected to the default wifi network.Once conected a user is notified and the user not can update his/her passcode. After a client side validation passcode is then send to the backend server in a secure channel. From the webservice at first the username and password are both checked and if matched the password is updated with the new code and the status is sent back to the device.If failed failed message is shown.

Once the user reaches this interface all the highlevel functionalities that the user being performed will be logged. And if the user had any access to a wifi network will be lost. In order to process the users request it needs to be connected in to default wifi network and the data will be communicated in a secure SSL/TLS tunnel for better security.

Once the user goes back from this interface app will disconnect from the default network and return to the Dashboard and re authentication is not required. User can select any asigned network and proceed.



You are connected now proceed

Figure 16: Account Screen WIFI-Connected

In order to perfor this task the app must connect back to the default wifi network. And when the user is loged to Dashboard default is removed. Once the user opens this screen in the background the app will again connect to the Default wifi network untill then the user is unabled to perform any task in this system.Once the app is connected with the Default wifi access point user can then change his/her pass code.

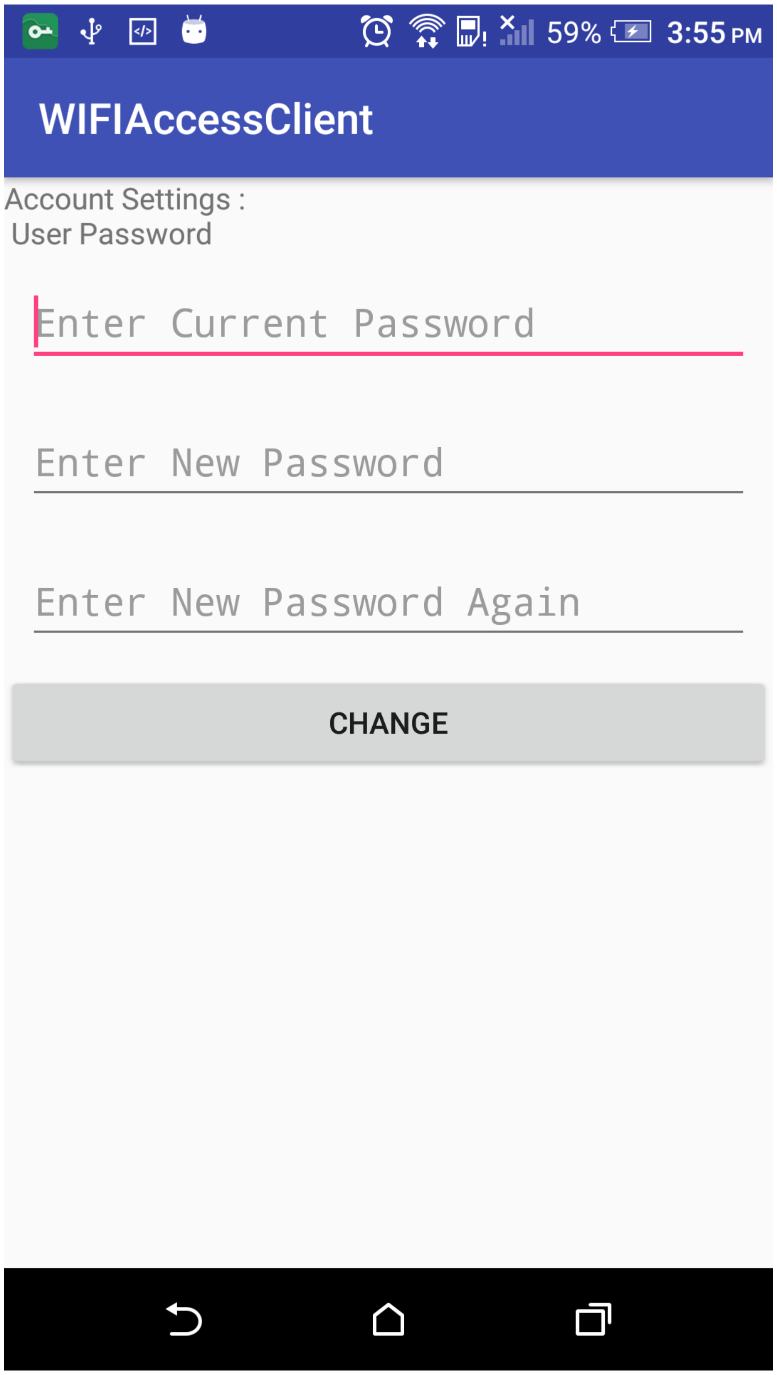


Figure 17: Modify Password

In order to change the pass code username is taken from the app itself but the password is required from the user. So as inputs current password and new password is required. To avoid mistakes in typing new passwords re type password is also added. A password has to be more than 8 charactors and anything less is tried this functionality will not work.

4.1.2 Admin

Once the user authentication is accepted if an only if the user has admin priviledges the user is then directed to the Admin Dashboard where he/she could do advance set of things like enrolling a user, view activity log etc. If the FaceID to be used the users face should be configured and registed with Kairos API.

This process is possible with this application. First the desired user should be selected by feeding the username of th respective user. Once validated the user is directed to a camera window where the face recognition functionalty is taking place. Same process just like in Verification is followed here, user needs to blink its eye so that this will capture the image of the user and then feed it back to server API and once a response is received from the server user is directed back again to the Admin Dashboard.This process looks so simple but the functionality behind this is not so simple.

First at the selection the when user feed the desired user's username or the email address a web request is called and get the verification, if found the user is allowed to capture and enroll or else will not proceed. Also to make the face assurate up to 8 faces can be used so that it would allow the recognition easier and better. Example if the user is enrolled with a full beared but after cuple of months the user decided to shave in this case an extra face to the exisiting face is required to be enrolled to avoid further issues.

The next important feature in Admin Dashboard is the web access from the android webview. With this admin user can logs in to the Server Dashboard from the same app and use admin actions like adding a new user adding a Wifi SSID asigning a WIFI account and view user Activities etc.This communication is also secured as the communication is also done through https connection.

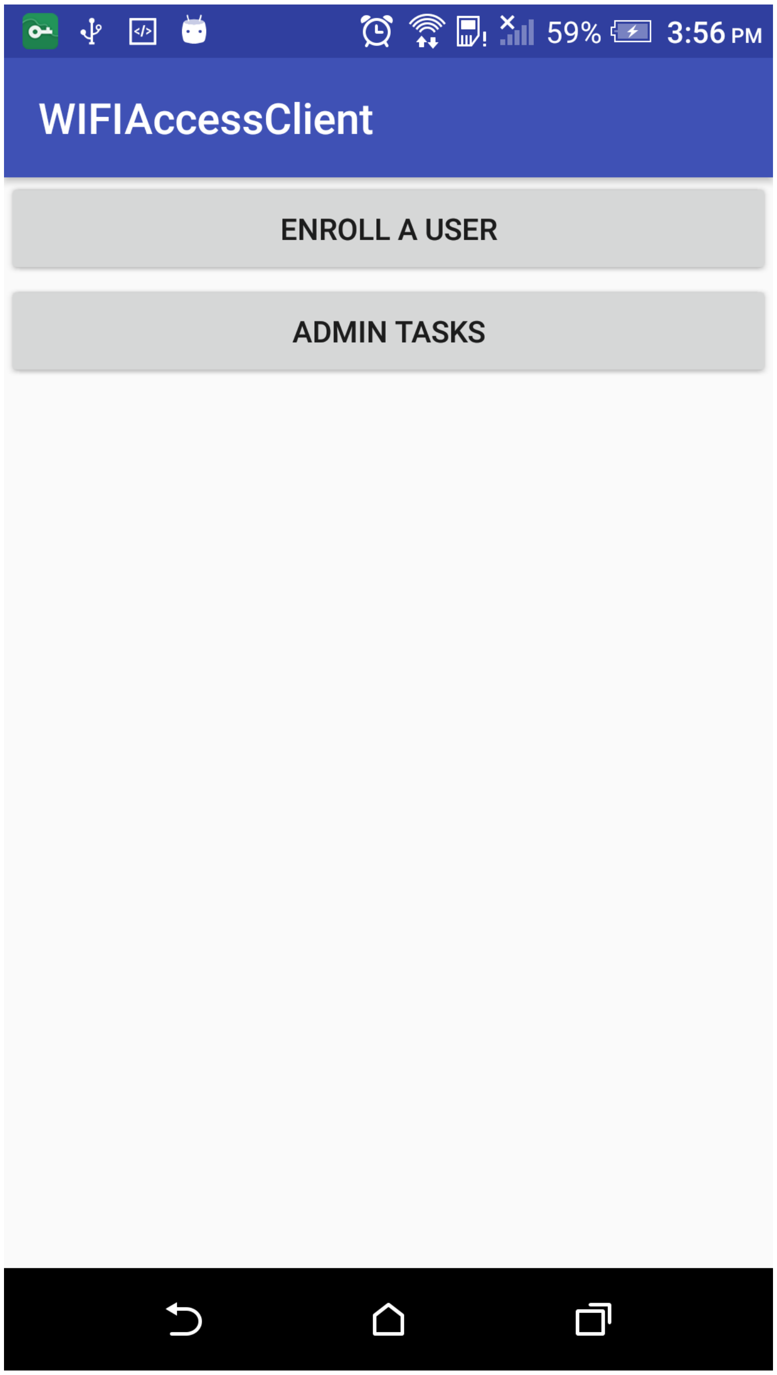
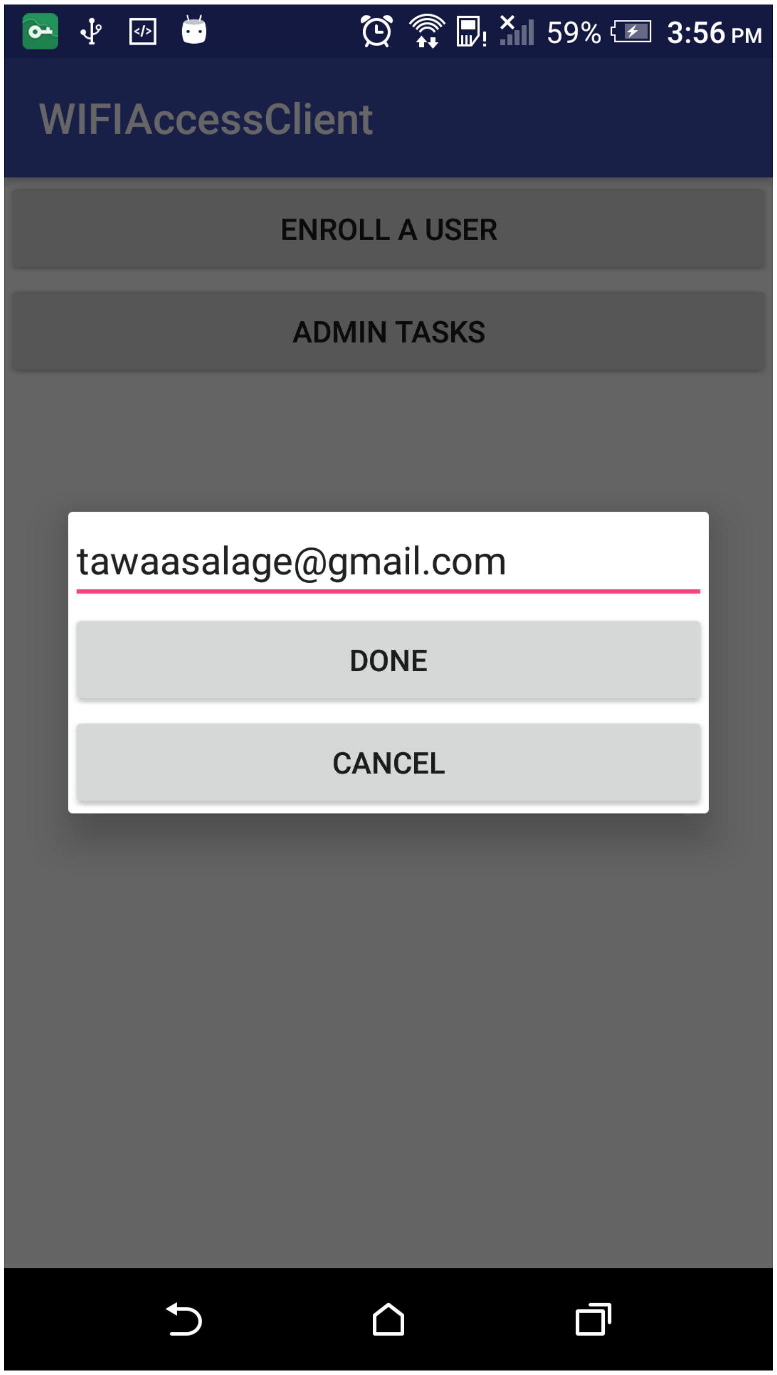


Figure 18: Admin Dashboard

Once the admin user tries to log in to the system the first screen the user sees after login procedure is the admin Dashboard. From this screen the admin is enabled to to administrative tasks. Enrolling newly added users in to the Kairos network and access Admin tasks module.



tawaasalage@gmail.com

Figure 19: Enroll User

In order to add a user's face ID in to our development which allows the user to use the two factor authentication this module is used. First the users email address which is email address need to be taken as the input. And once that is validated user is directed to Face recognition module here also the user needs to blink his/her eye to trigger the event. And the image is captured and send to the server for processing.And bas upon the reply if the user is successfully enrolled navigated back with a success message else the proper message is shown to the user.

4.1.3 Web

As I have mensioned above the development consists with two major parts. Front end android application and the back end web development. In this section how the back end is designed and what functionalities it provides are listed here. Mainly the back end is designed for admin tasks such as Adding Users, adding Wifi SSID's, asigning Wifi Accounts to users etc. And other than this is the background it holds an link to an API which comunicates with front end Android application in performing asigned tasks. And in this section all that is available is explained along with screenshots.

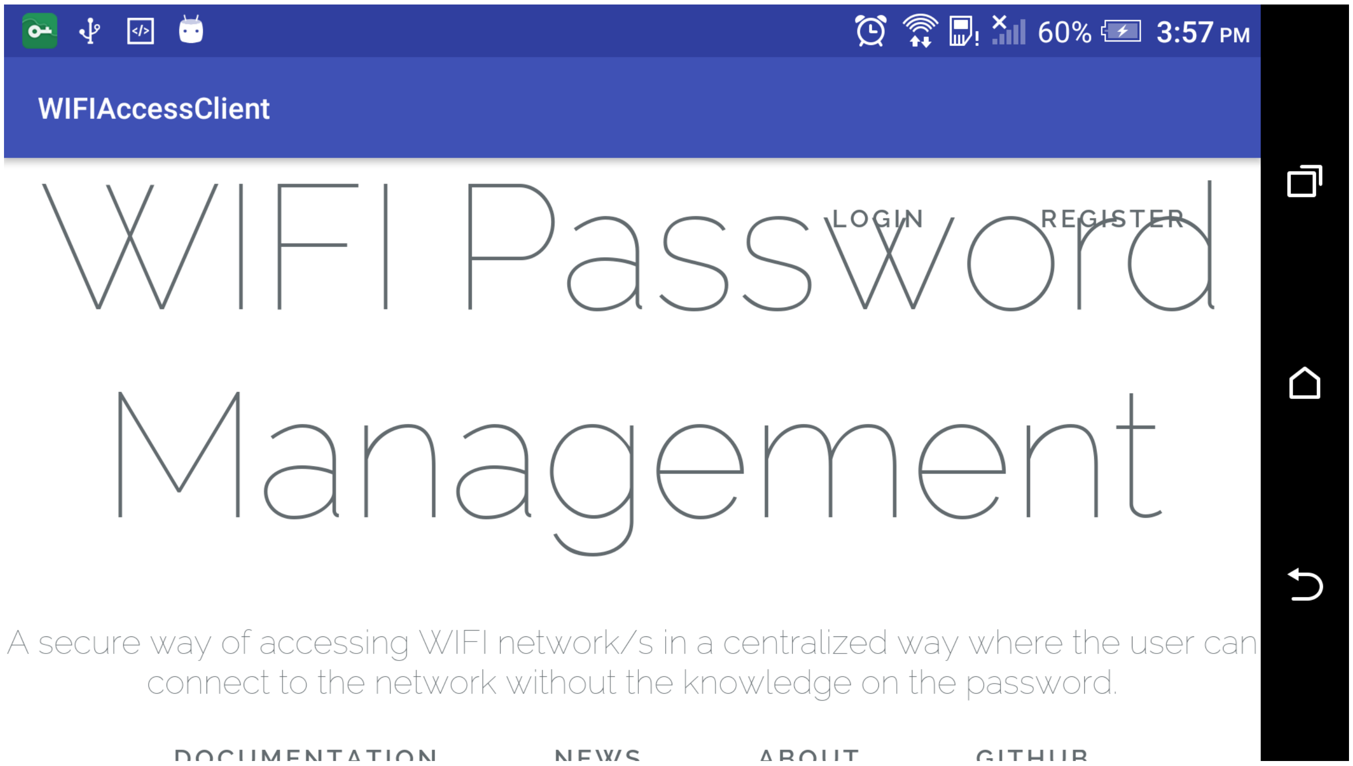

60%
3:57 PM
WIFIAccessClient
WIFI Password Management
LOGIN
REGISTER
A secure way of accessing WIFI network/s in a centralized way where the user can connect to the network without the knowledge on the password.
DOCUMENTATION
NEWS
ABOUT
GITHUB

Figure 20: Admin Tasks

Every function that can be accessed from the backend web module is possible from here.Which means the admin can log in to his account from here and then perform task like adding users,adding networks and asigning networks to users as well and viewing activity logs.

4.1.3.1 Admin Web Components

Login Interface:

This is the interface appears if the user tries to access the web development for the first time and as for the fact direct access is disabled if user tries to do so he/she will be directed to this screen. And this web module is only defined for the users with Admin priviledges. Also the user password is hashed and saved so even if the database is compromised the password is on no use. Once the proper credentials are used the user is directed to the user login screen.

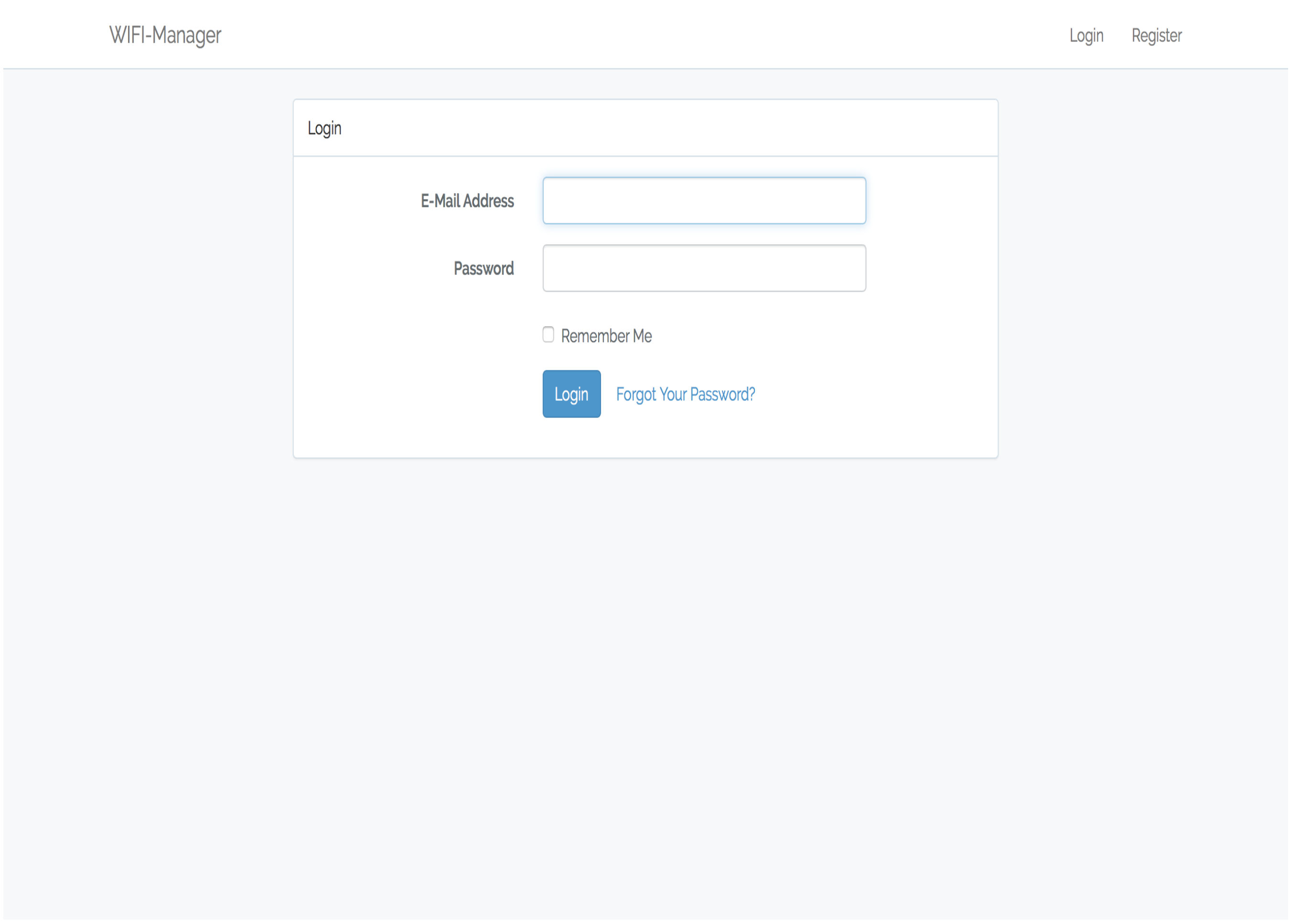


Figure 21: Login Screen

Login Middleware:

Current web development allows only the admin priviledge users to use, in doing so a small midleware script was written which validates the users admin status, if the user is an admin priviledge user user is directed to Dashboard and in case the user is a standard user no further actions and directed to the login interface.

Dashboard:

This is the place where the loged Admin user will see. As shown in the below Figure some set of high level functions are possible with this screen.

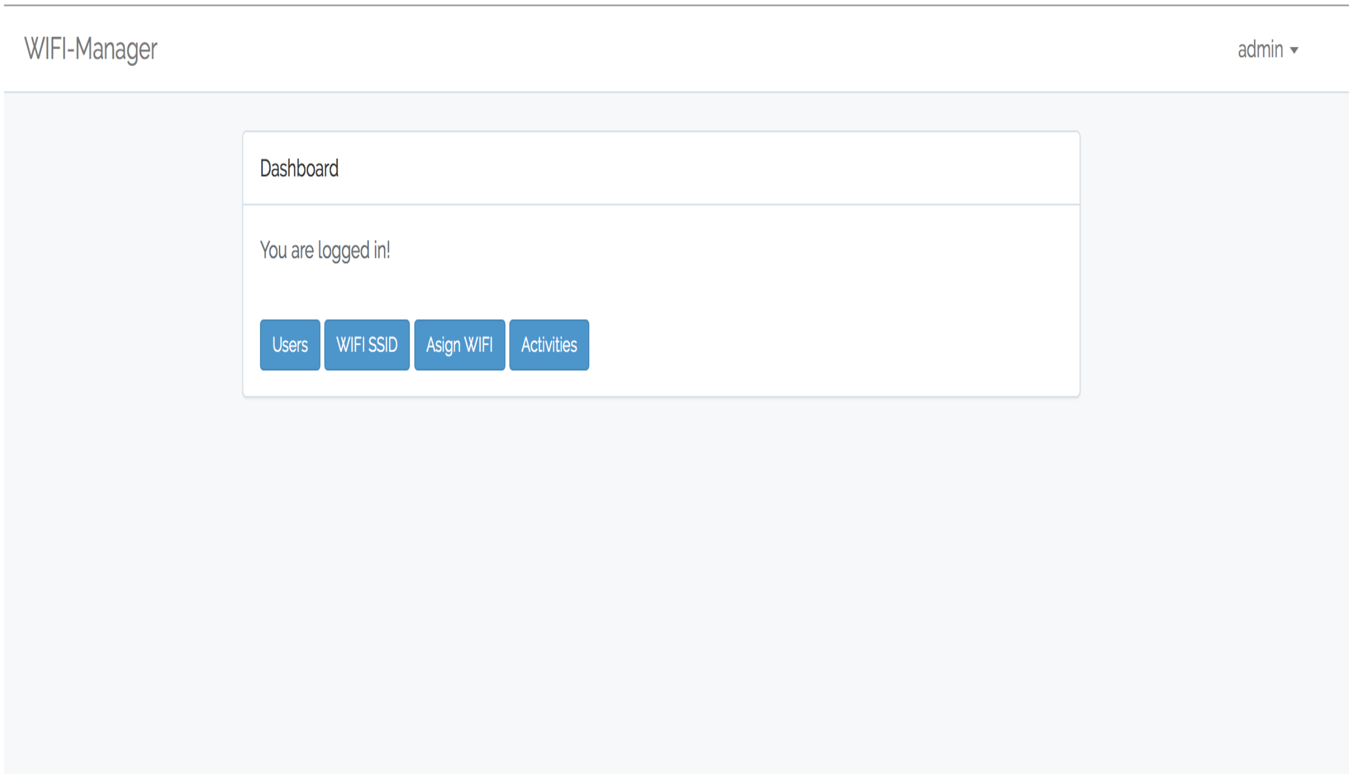


Figure 22: Dashboard Screen

Users:
Currently available users are listed in this screen. Additionally the Admin user can add modify and remove a user. When adding a user Name, email address and password is used as mandetory data and also it is also decided whether this user has two factor authentication or not. At the registration the password is taken in plain text and then hashed and stored in our database.

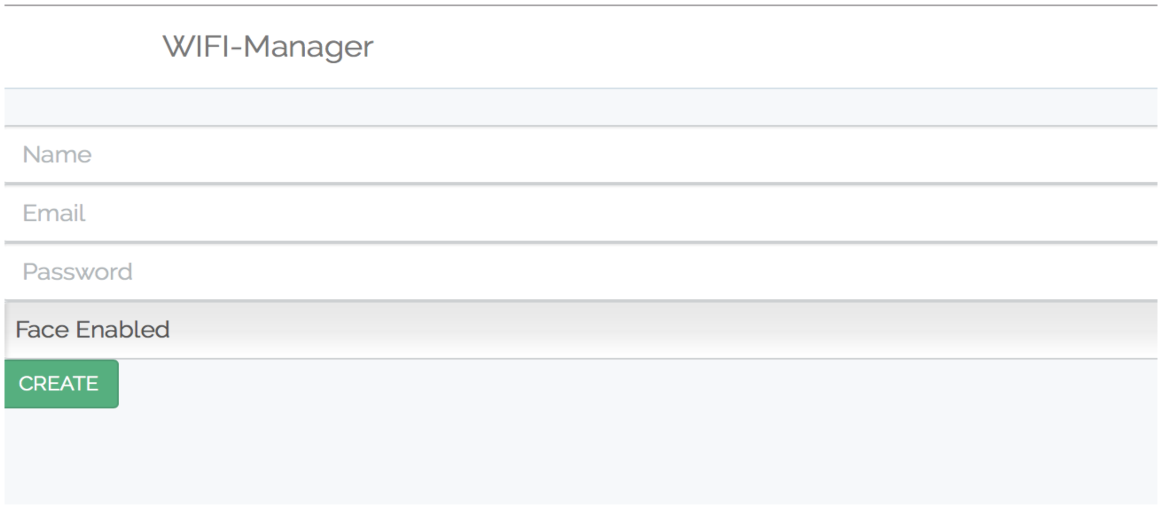
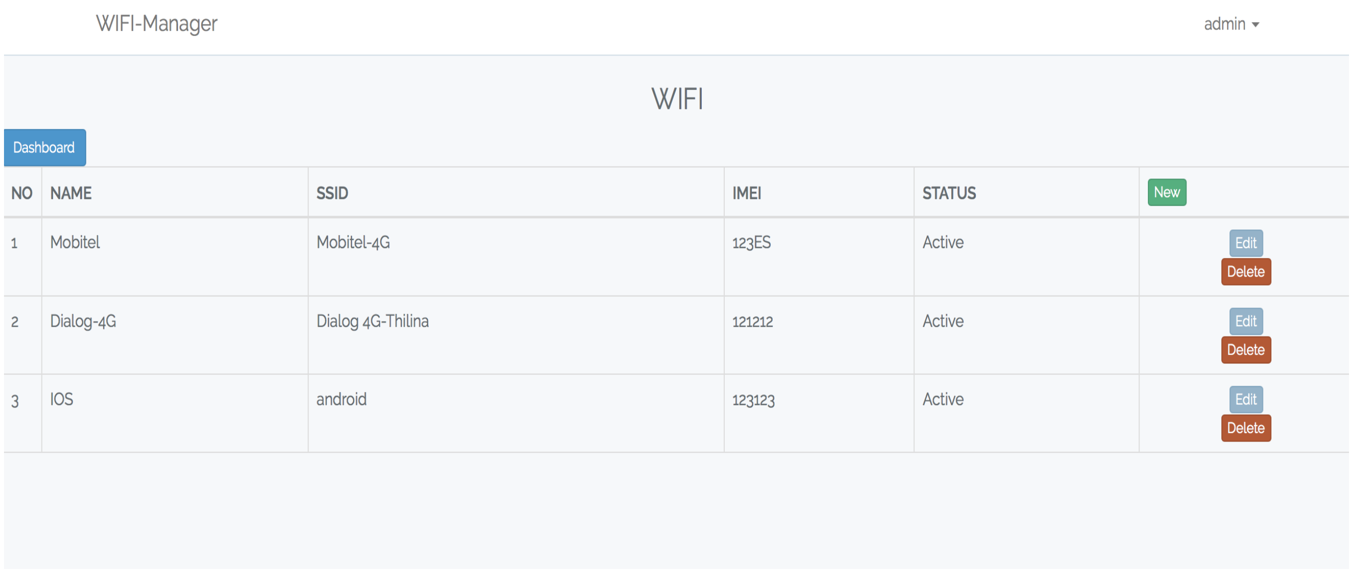


Figure 23: Add new User Screen

However in edit funcationality except for the email address the rest can be changed. Delete button will remove that respective record from the database and all these three actions are recoded as who did this action to what record etc. The records in Activity table can be viewed but cannot be removed from the web no such interface created for that purpose, however if you have access to the mysql server running on this pc it is possible to alter data again we'll be securing the mysql server with a custom secure password rather than using the defaults.

WIFI SSID:

This interface list down all the available active and inactive wireless Accesspoints. In this also Add new, Edit an existing or removing the accesspoint from the system is possible. Once the WIFI access point is deactivated already asigned accounts will not work from that point onwards.Actions performed on this interface is also logged in the activities table.

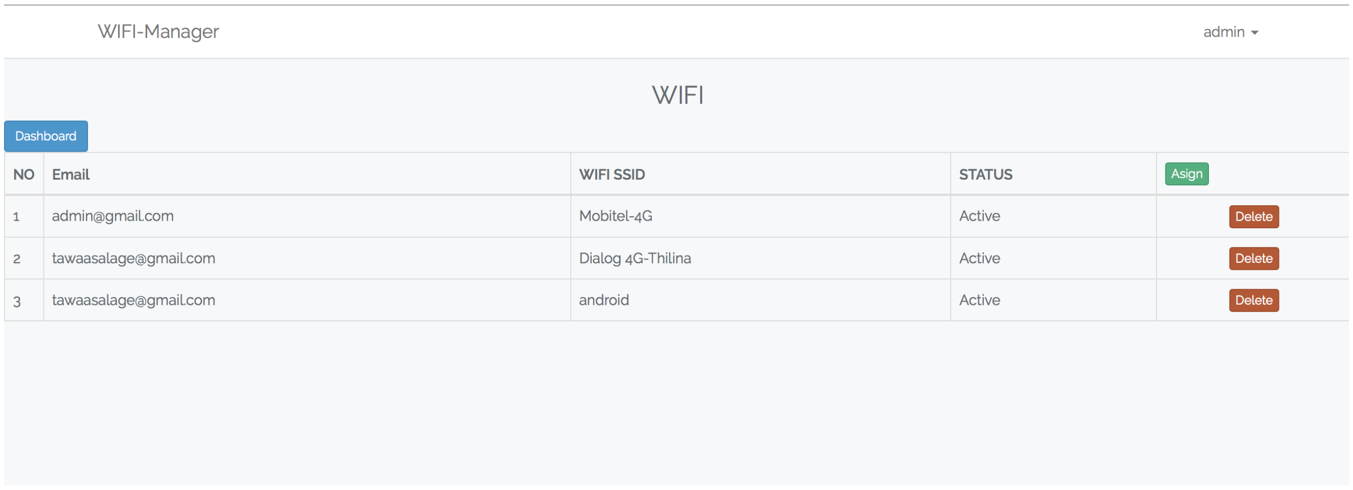


NO	NAME	SSID	IMEI	STATUS
1	Mobitel	Mobitel-4G	123ES	Active
2	Dialog-4G	Dialog 4G-Thilina	121212	Active
3	IOS	android	123123	Active

Figure 24: Add WIFI Screen

Asign WIFI:

Once a Wireless network is added to the system in order to get its work the network should be asigned to a user. This functionality is performed in this user interface.At first it will list down all the asigned users with their networks order by the user.When asigning a user to the network username and the SSID that needs to be shared is selected and also the status. Once this is done when the user tries to log in to the client application this record also will appear on that respective user's dashboard. In case of a mistake the user cannot edit but removal is allowed.Actions performed on this interface is also logged in the activities table.



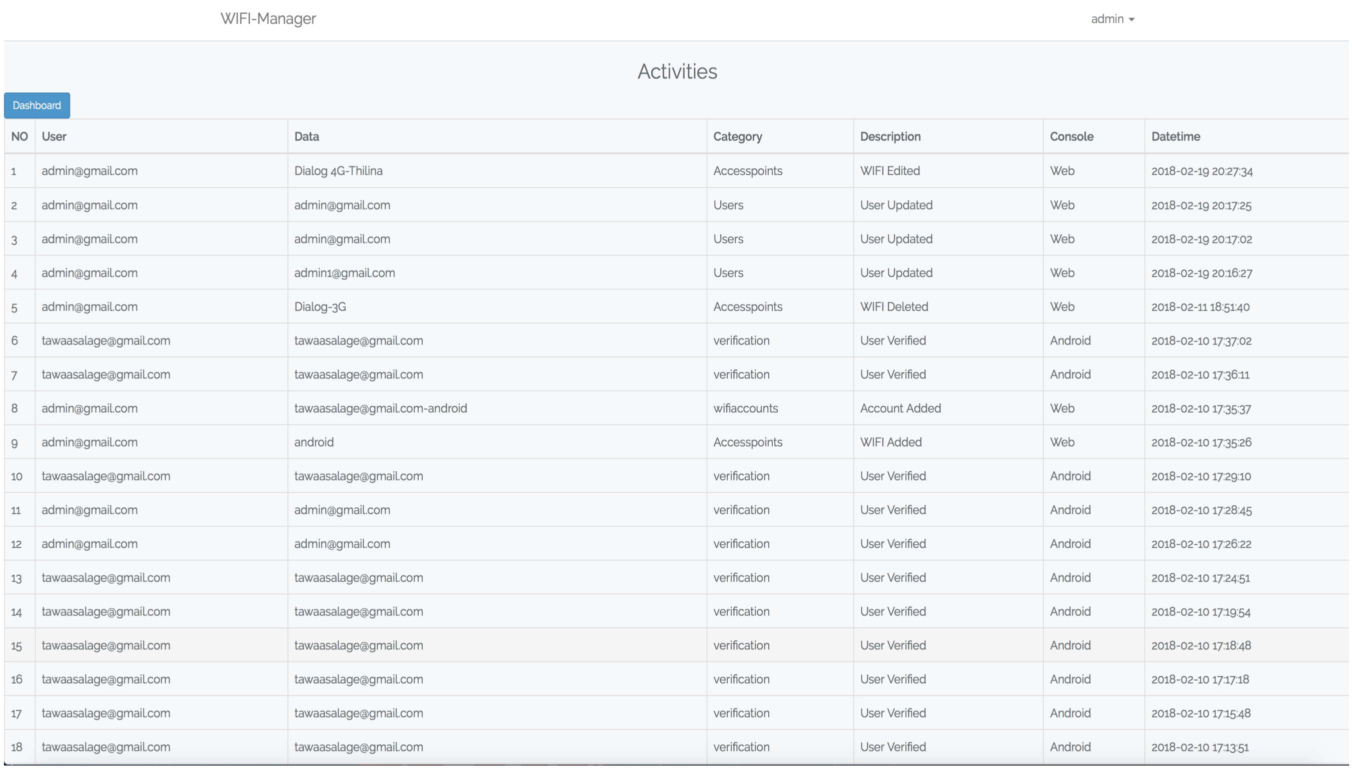
WIFI

NO	Email	WIFI SSID	STATUS	Asign
1	admin@gmail.com	Mobitel-4G	Active	Delete
2	tawaasalage@gmail.com	Dialog 4G-Thilina	Active	Delete
3	tawaasalage@gmail.com	android	Active	Delete

Figure 25: WIFI Asign Screen

Activities:

All the highlevel activities that any user does is recorded for reporting and for security purposes. This includes from Adding a user to removing a account inclidng web calls from Android device.



Activities

NO	User	Data	Category	Description	Console	Datetime
1	admin@gmail.com	Dialog 4G-Thilina	Accesspoints	WIFI Edited	Web	2018-02-19 20:27:34
2	admin@gmail.com	admin@gmail.com	Users	User Updated	Web	2018-02-19 20:17:25
3	admin@gmail.com	admin@gmail.com	Users	User Updated	Web	2018-02-19 20:17:02
4	admin@gmail.com	admin1@gmail.com	Users	User Updated	Web	2018-02-19 20:16:27
5	admin@gmail.com	Dialog-3G	Accesspoints	WIFI Deleted	Web	2018-02-11 18:51:40
6	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:37:02
7	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:36:11
8	admin@gmail.com	tawaasalage@gmail.com-android	wifiaccounts	Account Added	Web	2018-02-10 17:35:37
9	admin@gmail.com	android	Accesspoints	WIFI Added	Web	2018-02-10 17:35:26
10	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:29:10
11	admin@gmail.com	admin@gmail.com	verification	User Verified	Android	2018-02-10 17:28:45
12	admin@gmail.com	admin@gmail.com	verification	User Verified	Android	2018-02-10 17:26:22
13	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:24:51
14	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:19:54
15	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:18:48
16	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:17:18
17	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:15:48
18	tawaasalage@gmail.com	tawaasalage@gmail.com	verification	User Verified	Android	2018-02-10 17:13:51

Figure 26: Activities Screen

4.1.3.2 Web API

enroll:

This API is written to accept informations from the user and register/enroll the user in Kairos database. On success event a JsonObject is passed back to the caller stating the success or the failure. In order to call this webservice this service itself requires some parameters to be sent as a JSONObject. Once the required data is matched and validated a proper web call to Kairos API is generated.

verifyUser:

Used to verify the user with user credentials and if both matched a response with asigned active WIFI accounts are sent back as the response.

verify:

This act as the two factor authentication function. From device to face image is sent and from this api image is captured and compared with Kairos API to find a valid match. If a mathc is found then the accuracy level is checked. If it is above the threshold 0.6 we asume the result is accurate the the user is who he/sho is.A sensong in JSON is sent back to the caller.

verifyemail:

Simply returns the unique id of the email address. For anybody who looks at this perticular service might think this is bad very bad by knowing the username anyone can generate its primary key which is id. But for this development id is used only for face id so the things that can occur due to this is limited.

getActivities:

This webservice generates a JSON Array based on the date passed.An additional filter date is required as the parameter.

setActivityLog:

A function written which allows the system to log actions permored on sensitive data.

listWIFIpoints:

This API renders the WIFI accounts asigned to each user ID. Even if the url is called no harm as the password is encrypted using 128bit symmetric key which makes a malicious user to crack the password unless he/she grabs the private key.

4.2 Network Connection

Network connection in this development is something very essencial. Even the target application is fully designed to support wireless network access.So first the server should be fully equiped with the internet connectivity and as well as supporing direct access to services via a url. If we talk about the network configuration that we are providing for the Server, we are using a wifi based network for this whole process. But also in order to make things more secure we could combine two networks LAN based network which is connected to internet and a Wireless network and that would be connected to the LAN network in order to give mobile devices the access to the servers and by doing so it is certain that when the mobile is first connected to this wireless network the user will not be getting internet access.So these are the two ways that we could use networking to suit this development to make it efficient.

4.3 Security

In this development if the security is not been focused the validity is not there. And staring from small silly things to very complecated logics are required to be more focused on in designing this solution. First we need to avoid outsider attackes and once that is done we need to focus more on inside attacks which is more common in corporate world. So basic security principles CIA or else known as Confidencialty Integrity and Accessibilty is pretty much covered in this development.Confidenciality, this means we need to protect sensitive data from intruders like hackers or any third party entity.

4.3.1 Database Security

According to the solution the WIFI ssid password is something really sensitive.If this can be accessible the whole objective of the development is no more, infact this can be mensioned as one of the top assest we try to protect in this development. As we know the storage medium we used to store these password is MYSql database. But these pass codes are kept in plaintext anybody who has the access to the current database has the ability to read this password. This goes same with the user password. But in order to make things more secure encryption is introduced in dealing with passwords.

As for the encryption we used 128bit symmatric based encryption which also called as private key cryptography. Data is encrypted and stored in mysql database making it really hard without the key it is not possible to decript the password. But if the user has fully access to the Server machine the attacker can grab the logic behind this development and re use the code for his own use.

4.3.2 Physical Security

So the other dangerous thing that we need to be aware of is that physical security of the server machine. As commonly known if the server machine is fully compromised we have no options and the attacker could do things he want starting from reseting mysql password to stealing source code.

So with this it is very much clear how important the physical security is. So inorder to make the server tight there are few things I'm going to follow.One change default ports. There are common ports that people are more familier with like 80 for http 443 for https 21 for ftp etc. So if we were unable to change these ports any intruder will gess and try to gain access to the system via these well know ports. So the first thing must follow is that to chnge all important common ports.

Avoid giving the root access directly is something that we need to avoid. Incase the root priviledge is need the user should be able to authenticate and gain access.So with this any attacker who gains the access to the server via some method will not have root priviledges which would not permit to do additional administrative tasks.

Using a complex user credentials on the server is another best thing that we need to practive.The password on the server should be a complex one should not be weak.And even if the credentials on the application is set accurate but with weak security in the server is detected we are making a huge mistake here.So this is another major point we need to look into in making this development a better and secure solution.

4.3.3 WIFI Security

In general WIFI technology wireless adapter has two modes access point mode and monitor mode. In accesspoint mode the adpter act as a bridge to a wireless network where the user who uses this device tries to connect with a network,But monitor mode is something else when this is active basically what ever the data communicated will be grabbed and saved in the device. So if the communication is not encrypted the attacker will get hold all the data the user used in plain text which is something we need to avoid. So while the Accesspoint is in monitor mode it is simply not possible to connect with anyother accesspoint. This mode is only for people who likes to sniff around. So with this any user can see or sniff data from other wifi connected nodes. So after a while they could dump files and monitor for sensitive data.

As I have mensioned in the above step there is only one way that is possible for us to solve this issue. That is via encryption that already we are following.For the development we are using HTTPS based communication for webservices and as well as the server direct access.So with this the data shared across wifi ssid is not readable.For this development I have adopted HTTPS communication when communicating with the server from Android device.

The main idea of this whole project is to allow a user to easily connect and access a wifi access point with out revealing the password of it. As any one would know by this time at first when the application is launched for the first time default WIFI network is considered as the initial point of access. As I have explained before we have two options when defining the first access point, one use a default wifi with the server and internet connectivity should be available on this in order to provide the server to access Kairos servers for face authentication purposes. Then we have the other step which is to connect two different networks together one with internet connectivity and one with out.Ther server will be working on LAN with internet connectivity where as the android nodes will be connected to the server via a wireless network with out any internet connectivity.So in this case we no longer have to wory about the security in the first wifi access point.We can simply give full access to the default wifi ssid and make it a public one. But for this development purpose we asume we are going to use the method one which is one network for all. So we need to avoid the user to gain the default network access for a long period of time which would cause us problems.So the best thing we should do is that keep the user connected to the default network for the less time. And avoid the user doing things that would keep them on using the same network for a longer period. So as precautions when starting the app in order to make things secure we are asking the user to make this app administrator. So with this we avoid the user to force close the app or delete the app once the user is connected to the network which would stop all the background works that will lead the user to access default or connected networks for any duration they want. If the user set this application as the administrator the user cannot perform the tasks above. In order to do so the user has to deactivate the admin mode. But when the user does that our app will be open and the app will clear all the connected networks from the device which makes it really hard for the user to play with this system. So these are the ways that I looked at the security side of this development.

Another point I found in dealing with these technologies is that the way of handling password hashing. Unlike generic methods with new functionality whixh is provided by Laravel it makes insanely easy to secure user passwords. But this same concepts cannot be appied for WIFI SSID passwords. But with this added feature in directly it affects the security in a good way.

4.4 Exception Handling

Exception handling is one of the Key components we look forward when it comes to developing. Because a simple mistake would lead the whole development in vein. Hence we cannot allow these simple mistakes to make our development less sure and buggy.So the Exception handle is comes in handy. Most common issues we comes accross in this development is network related issues. If the connections get time out and if something hapence with json etc. So the correct Exception handlers are put in place. And further more Firebase Error reporting package is also in build so that if incase the app get crashed we know where to look for as the API is fully equiped in maintaining logging and sharing the log cats with the developers.

Chapter 5

5 Installation

5.1 Server

Once the application is ready for production first the server is reqired to be hosted. Currently the backend is designed in php so the server machine should equiped with apache along with mysql as the database. As this development supports https communication port 443 is also can be used but this is not mandetory.And ther server ip address should be a static ip where it should not change once the server restarts. And this server is required to be connected with the default wifi ssid that would allow the client application to connect. If this application is going to be used in large organizations we cannot rely on just singe accesspoint.What we could do is that we can simply create SSID with the same configurations and place covering all the possible points lets say floor wise. If needed floor wise we could maintain client builds so that we don't have to use the same configuration in each floor. Once this is done lets move to the client side.

5.2 Client

A client android build should be published on google play so that the user can install the app for the first time from google play.Once installed in order to use the functionality the user has to enable administrator mode. App will then tries to connect with the default server once connected login screen will be prompted. From there the given user credentials needs to be used in order to gain access to the system.If the user have no such combination a free account can be created with the help of the administrator. For guest usage a dedicated username and password can be provided and so that the guest users are not required to get themselfs an account. So once the user logs in from the Dashboard select the desired wifi SSID and enjoy.

Chapter 6

6 Evaluation

In a development that main thing is the usability of the final outcome which is the final product in our case the Software which allows the users to maintain manage wifi access points in a proper distributed way.

So inorder to prove that this actually works we cannot keep it to our selfs and test. We need to share this others so that only we know whether this solution is perfect for them or not or do we need to modify it. This is more like a beta test. There is one thing when it comes to developments, even the huge organizations like apple & windows are not wait after releasing one version.When people test it they identify the strength and weaknesses which allows them to do more in the next release or if a major bug is found they could take precautions.

So in this case at first the development was implemented at home and as well as at my office. Based on the requests and issue reports and suggestions from users lead me to modify the app and come up with a good final build.So from this section the test results and findings will be briefed along with few reported issues.

6.1 Phase 1

Tests at Home environment:

At home environment two wifi connections from Dialog and as well as SLT which allows over 5 mobiles and 3 other devices to connect to the network. As a result 4 simple user accounts were created and 2 wifi accounts provided internet access to users.In this case 2 phones to use Dialog and 2 other devices to connect with SLT. One mobile device was not connected as it was running IOS and due to the fact that the current development only targets Android IOS devices were ignored.

From the server two factor auth was disabled for couple of days to make this easy for us and made the timeout for 1 hour from backend for testing then slowly enabled the two factor authentication for family members. There were issues at first like connection drop and code issues but was able to fix things quickly and updated their phones so for a week time the test went perfectly. So then based on the reviews had from users with out two factor auth it is way too easy for them to connect to wifi.In the mean time for beter testing the password was changed multiple times in wifi routers as well to get the real benifits. So back at home things were going well and with that success I moved testing at a business environment.

6.1.1 Results

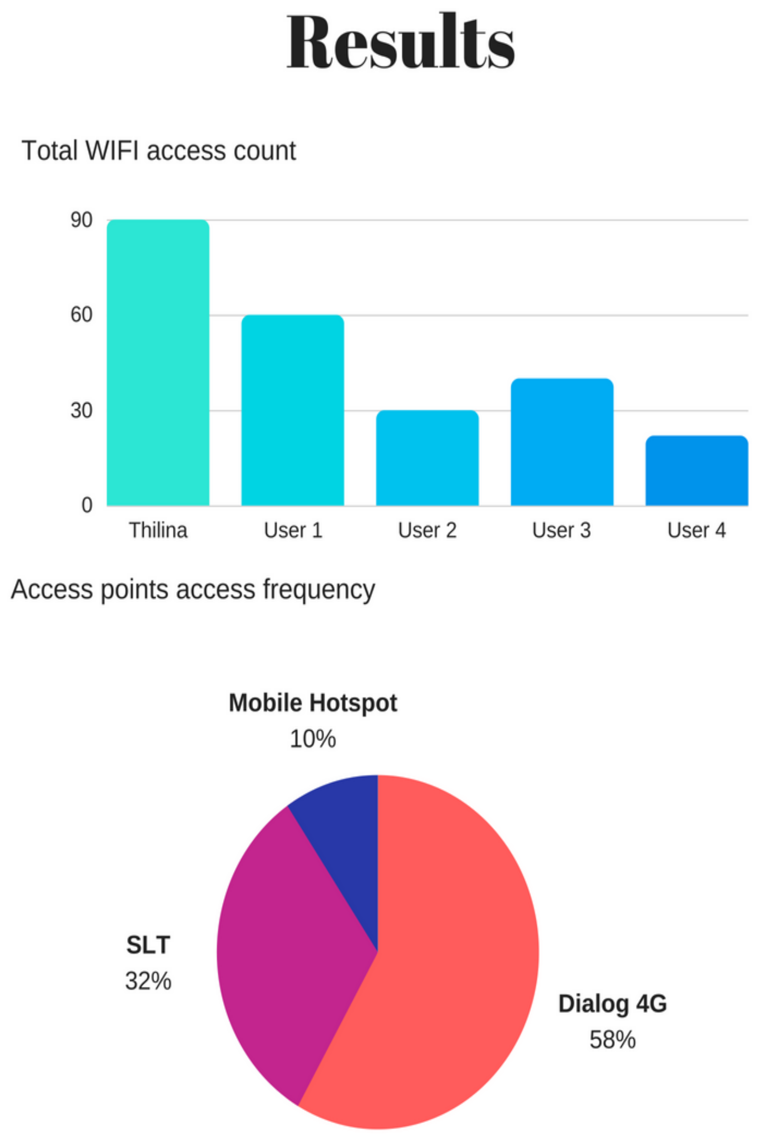


Figure 27: Test Result Report

Based on the data aquired over 2 weeks the above report was generated. Total WIFI access count section indicates the total number of times that a user has tried to connect with the solution. Note that the names used are all imaginary. Thilina has accessed the 90 times user1 60times,user2 30 time and user3 40time and finally user4 20 times. Alltogether total of 142 times app access is recorded. The next section indicates that most commanly accessed accesspoints. Dialog 4G AP is been accessed 141 times and SLT router being accessed for a count of 23 and then the SLT AP has being used for 78 times.

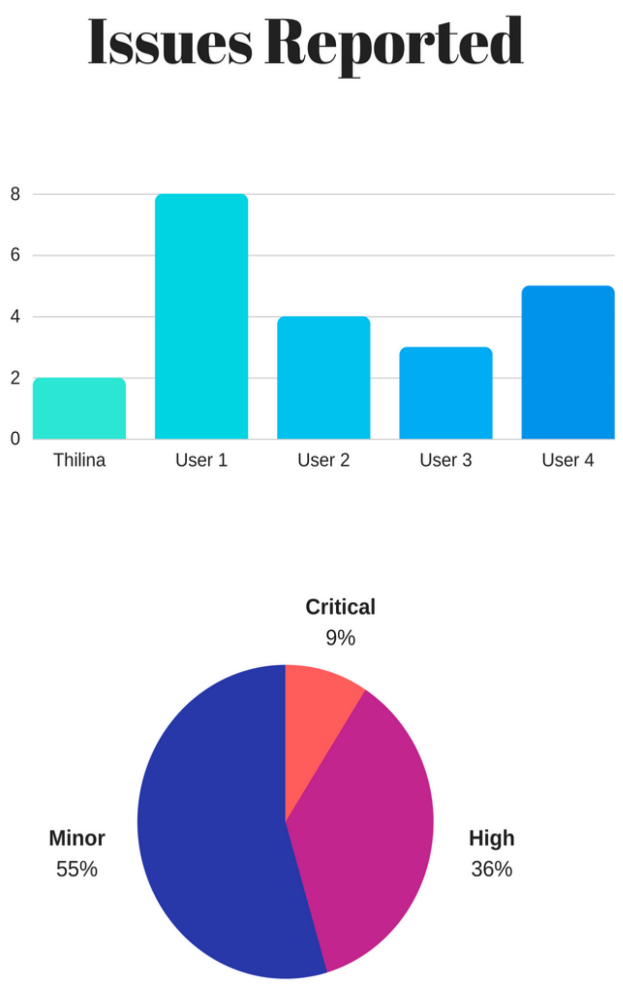


Figure 28: Issues Reported

The above diagram illustrates the issues encountered during the testing in phase one. Most of the issues reported by users were minor but few critical & high rated issues also were reported and then patched a new version by fixing the issue. Overall of 22 issues were reported and from that count 12 minor 8 high risk and the remaining 2 critical issues were found.

Critical Issues reported:

- **Once Connected** if the app is forced closed user will always connect with the WIFI network
- **Once Connected** if the app is deleted user will always connect with the WIFI network

6.2 Phase 2

Tests At Office environment:

As the next phase for this development the implementation was done at a Office enviroment to give this development a better chance in future. As the office was already having a system running with face recognition the images took at the time of registration was used here in this development too so separate aditional process was not needed. At Office environment a ubuntu server running ubuntu 17 server that can be accessible through a wired LAN and as well though wifi network was used for the implementation after requesting permissions from management permission. As their is already generated a static ip for that on the android side the ip was changed in the code to take this static ip and allowed connectivity from the default router. At office 2 internet connections from SLT & Dialog along with wireless connectivity is available. So in this case created 3 networks 2 from accesspoints and one as a hotspot for testing purposes.

Users from the office tried stuff on there phones not like back at home they found several ways of by passing the scurity checks that i have being placing. Force closing the app uninstalling the app , exit once the device is connected to the default network thing like that. With their comments received from experienced developers looked more deeper in to the security side of this development as well as the best practices that could use in order to improve the quality.As a result a solution for uninstall and forced close issue was found and a patch was released besed on the reported issues.

And regarding the two factor auth never received any issues as they are very much familiar with the process. And another comments from the team is that to increase the connectivity duration as 1 hour limit sometime annoys. So as a result the duration was enabled as a function to set the time duration from backend so that if a time is set in hours it will reset in the selected hour else 0 is selected never but if the app is re launched the whole process is to repeated.

6.2.1 Results

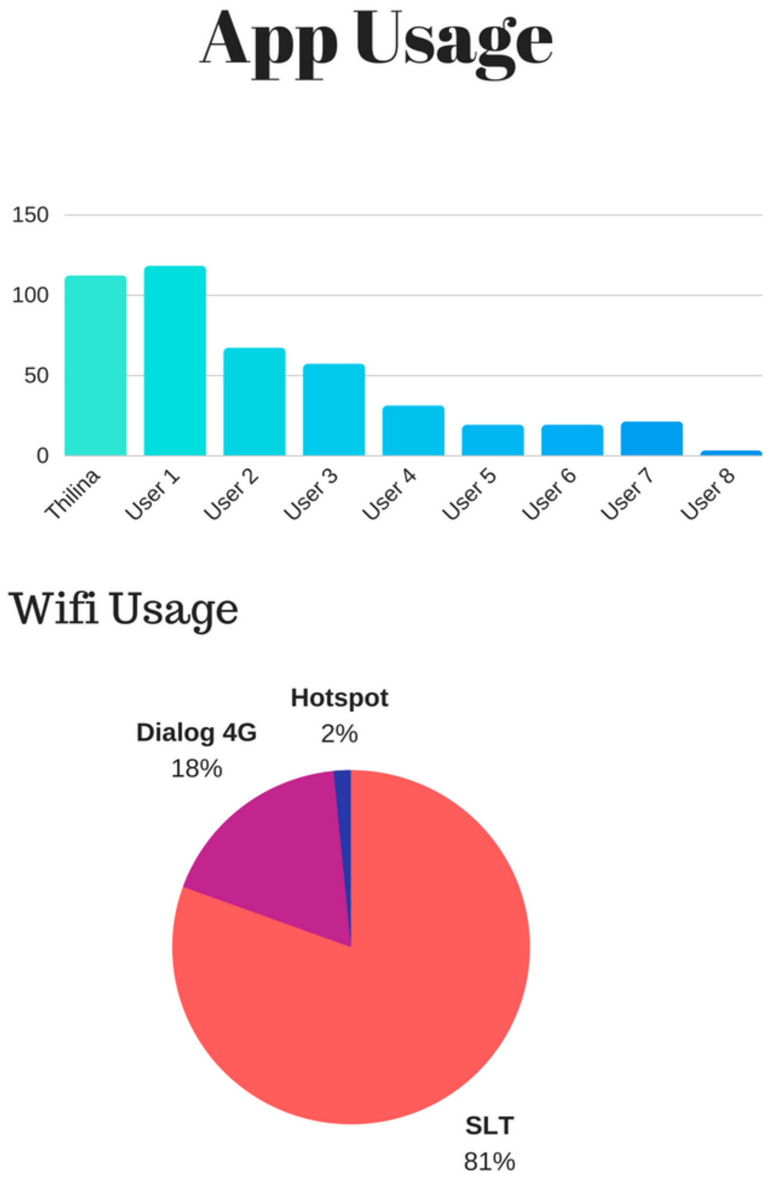


Figure 29: Office Test Results

Once the solution implemented perfectly on the home environment the next phase of implementation was done at a office live environment where over 8 users daily for a period of 2 weeks the solution was used to connect with the network and above report illustrates than in graphs.

6.3 Effectiveness

The solution was worked perfectly for what it was build to accomplish, to minimize admin enrolment in password management and provide ease and centralize way of accessing wifi related credentials in a secure manner. Based on the reviews received from many users the solution provides them the basic connectivity and track their usage and ability to change credencials time to time with out notifying the user was really interesting.

6.4 Performance Results

Performance		
Process	AVG time in (s)	Error rate
Login Using user credentials	2	0
Face authentication (Slow)	8	2/10
Face authentication (Medium)	6	2/10
Face authentication (Fast)	4	2/10
Connect to a prefered Network	6	1/10
Connect to default Wifi Network	6	2/10

6.4.1 Description

The above table was generated based on the results aquired from users who accessed the application overall results for this development was positive. Please note that Slow,Medium &Fast is an indication to the network strength.

Chapter 7

7 Conclusion and Obstacles

7.1 Obstacles

Although I have had a good experience with the current development I must mention that there were some instances where I had face some troubles and there are things I cannot work out with this release but later for sure I could find a good methodology to overcome the situation. Even though I managed to find a way to deal with face spoofing with an image I cannot do this for videos. The logic I have applied for the development will not overcome spoofing with a video although I have managed to find a way dealing with this I will not be using this for the moment. If I do it will directly affect the performance and will take more time based upon the network strength. I need to use 2 separate API to overcome this.

So far this has been the only obstacle I see in this project. And although I have not purchased the SDK from Kairos but the free version is more than enough for me to implement this system.

7.2 Limitations

Currently with free Karos API allows only upto 30 API request per minutes which is more that enough for a small set of users. But if this is to be used in a bigger environment a paid version is required.Other than that free service provides all the features in Paid version. In order to use this functionality the user must aproach near the default wifi network to gain the access to the system. So meaning access from home mecanism will not work here. If incase we host our back end in a remote server and through data connection provide access just like other applications like facebook,whatsapp then we could share password from anywhere with out a location restrictions. But if we do that the places where we might need to protect tends to increase and the real value of using a wifi connection will not be there.After all in most of the cases the main use of the wifi network is to get internet access but if you have the same connection via data connection what is the point. And also if you are asigned to a network simply clicking on it will not work.You need be physically available near the accesspoint or else the app will not connect with the accesspoint. So I see a limitation in availability. But this will not be a issue as this is beyond our basic requirements. What we need is to get access to the networks when we want inside the organisation.

Chapter 8

8 Future work

8.1 Cross Platform Development

Current version supports only Android platform but the idea is to enhance this development to support other devices such as Laptops & Iphones etc. Also from the next phase most commonly used platforms windows & linux is on target. For this to be worked out only the front end application is required to be designed as backend can be re used in both developments.

8.2 Spoof Detection

As mensioned above this development is weak againts video spoofing. In more detail the user is able to by pass the two factor verification by using a video of the user that blink his/her eye. Even though a working fix is found a API for this still not reliable due to its lack of performance and high network usage.So in future I might target this issue clearly and find a fix and integrate that in to the solution in making this a more secured application that the current build.Plus I'll be hoping to get new suggessions from the users and based on their reviews and comments integrate additional components in making this app not just an app that works but a usable app with benifits.

8.3 Security

As this solution is providing security directly to wifi networks it would be better to focus more on its weakness and try improving it until a stable development is up and running. So with this several key areas will be addressed in the next phase of the development.

8.3.1 Solution to Evil Twin attack

Also with the next step will be to find a better solution to prevent Evil Twin attack. Meaning a malicious attacker places his/her own wifi access point and sniff legitimate users data. So if the solution is going to be up and alive in big organizations this will be something to look at.

8.3.2 Alternative to the Symmetric Key

Wifi passwords in the current development is stored after encrypting with a symmetric key but it would be wise to apply something more advance like elliptic curve cryptography to give more power with high performance. This will also be looked in the next phase.

8.4 Location Base Connection

Current development does not tell the user what is the closest wifi ssid on range. but as an extra feature to improve the quality of this development wifi access points locations with latitude and longitude will be recorded and based on the users location nearest access points will be shown so that the user can easily pick the nearest one which would give them better connectivity.

9 References

References

[1] OTP RADIUS SYSTEM
Your Bibliography: Softlock.net. (2017). OTP Radius System. [online] Available at: http://www.softlock.net/Solutions/Solution-Details/2005?OTP-Radius-System [Accessed 24 Oct. 2017]

[2] VALIDCARD - BIOMETRIC OTP CARD
Your Bibliography: YouTube. (2017). VALIDcard - Biometric OTP Card. [online] Available at: https://www.youtube.com/watch?v=7IrNQM8G0fg [Accessed 24 Oct. 2017].

[3] XIRRUS ENTERPRISE SOLUTIONS
Your Bibliography: Xirrus. (2017). Xirrus Enterprise Solutions. [online] Available at: https://www.xirrus.com/enterprise/ [Accessed 24 Oct. 2017].

[4] IRISHIELD SERIES
Your Bibliography: Iris Scanner | Iris Biometrics Technology | Iris Recognition. (2017). IriShield Series. [online] Available at: http://www.iritech.com/products/hardware/irishield%E2%84%A2-series [Accessed 24 Oct. 2017].

[5] PASSWORD MANAGERS: ATTACKS AND DEFENSES: RETRIEVED FORM
Your Bibliography: Password Managers: Attacks and Defenses: Retrieved form. (2017). [ebook] Available at: https://crypto.stanford.edu/ dabo/papers/pwdmgrBrowser.pdf [Accessed 24 Oct. 2017].

[6] A KEY MANAGEMENT SCHEME FOR WIRELESS SENSOR NETWORKS USING DEPLOYMENT KNOWLEDGE
Your Bibliography: Anon, (2017). A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge. [online] Available at: http://www.ecs.syr.edu/research/SensorFusionLab/Downloads/Jing%20Deng/key_infocom04.pdf [Accessed 24 Oct. 2017].

[7] CITE A WEBSITE - CITE THIS FOR ME
Your Bibliography: Theseus.fi. (2017). Cite a Website - Cite This For Me. [online] Available at: https://www.theseus.fi/bitstream/handle/10024/95041/thesis_Opikhalov.pdf.[Accessed 24 Oct. 2017]

[8] THE BEGINNINGS AND HISTORY OF RADIUS
Your Bibliography: The Beginnings and History of RADIUS. (2017). [ebook] Available at: https://www.interlinknetworks.com/app_notes/History%20of%20RADIUS.pdf [Accessed 24 Oct. 2017].

[9] FACE RECOGNITION: A LITERATURE SURVEY
Your Bibliography: Face Recognition: A Literature Survey. (2017). [ebook] Available at: http://www.face-rec.org/interesting-papers/general /zhao00face.pdf [Accessed 24 Oct. 2017].

[10] ANDROID APPLICATION FOR FACE RECOGNITION
Your Bibliography: Android application for Face Recognition. (2017). [ebook] Available at: https://ri.iut-info.univ-lille1.fr/2013-Avril/Herning/LETUPE%20Rapport.pdf [Accessed 24 Oct. 2017].

[11] FACE RECOGNITION IN MOBILE DEVICES
Your Bibliography: Face Recognition in Mobile Devices. (2017). [ebook] Available at: http://web.eecs.umich.edu/ silvio/teaching/EECS598_2010/final_report/ Aditya_Srujan.pdf [Accessed 24 Oct. 2017].

[12] ART vs Dalvik
Your Bibliography: introducing the new Android runtime in KitKat. (2017). [online] Available at: https://infinum.co/the-capsized-eight/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat [Accessed 24 Oct. 2017].

[13] HTTPS with MAMP
Your Bibliography: How to get MAMP working with HTTPS. [online] Available at: https://gist.github.com/jfloff/5138826 [Accessed 24 Oct. 2017].

[14] LARAVEL PHP FRAMEWORK
Your Bibliography: ADVANTAGES OF USING LARAVEL PHP FRAMEWORK. [online] Available at: https://belitsoft.com/laravel-development-services/10-benefits-using-laravel-php-framework [Accessed 24 Oct. 2017].

[15] Android Networking
Your Bibliography: A series of articles digging in the Android networking libraries. [online] Available at: https://medium.com/@sotti/android-networking-ii-okhttp-retrofit-moshi-and-picasso-c381f6c0efd8 [Accessed 24 Oct. 2017].

[16] Overview of Android architecture
Your Bibliography: Overview of Android architecture [online] Available at: http://simpledeveloper.com/android-architecture/ [Accessed Feb. 2018].

[17] The Life of an APK
Your Bibliography: The Life of an APK [online] Available at: https://www.anandtech.com/show/8231/a-closer-look-at-android -runtime-art-in-android-l [Accessed Feb. 2018].

[18] Choosing AR or Dalvik
Your Bibliography: Choosing AR or Dalvik [online]
Available at: https://infinum.co/the-capsized-eight/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat [Accessed Feb. 2018].

[19] Client side code security
Your Bibliography: Choosing AR or Dalvik [online]
Available at: https://www.synopsys.com/blogs/software-security/android-security-best-practices/ [Accessed Feb. 2018].

[20] Overview of Android architecture
Your Bibliography: Overview of Android architecture [online]
Available at: https://www.kairos.com/blog/face-recognition-kairos-vs-microsoft-vs-google-vs-amazon-vs-opencv/ [Accessed Feb. 2018].

[21] Android ORM Performance
Your Bibliography: Android ORM Performance [online]
Available at: http://greenrobot.org/android/android-orm-performance-2016/, [Accessed Feb. 2018].

[22] Manage WIFI Passwords
Your Bibliography: Manage WIFI Passwords Article [online]
Available at: https://community.spiceworks.com/topic/285814-how-do-you-manage-your-wifi-passwords, [Accessed Feb. 2018].

[23] WPA2-Enterprise
Your Bibliography: Configuring RADIUS Authentication with WPA2-Enterprise [online]
Available at: https://documentation.meraki.com/MR/Encryption_and_Authentication/Configuring_RADIUS_Authentication_with_WPA2-Enterprise, [Accessed Feb. 2018].

[24] Google Vision API
Your Bibliography: Google Vision API: Image Analysis as a Service [online]
Available at: https://cloudacademy.com/blog/google-vision-api-image-analysis/ [Accessed Feb. 2018].

[25] Java
Your Bibliography: What Is Java [online]
Available at: https://www.thoughtco.com/what-is-java-2034117 [Accessed Feb. 2018].

[26] Java
Your Bibliography: A Review Paper on Face Recognition Techniques
Available at: https://www.researchgate.net/publication/270583005_A_Review_Paper_on_Face_Recognition_Techniques [Accessed July. 2018].