DEVELOPMENT AND IMPLEMENTATION OF A NEW ANDROID SOCIAL MEDIA APPLICATION

by

Bright Opoku

Comfort Osei

Jerold Baidoo

Benjamin Arhen

A Thesis submitted to the Department of Computer Science,

Christian Service University College

in partial fulfilment of the requirements for the degree of

BSc. Computer Science Faculty of Health and Applied Sciences

DECLARATION

I hereby declare that this submission is my own work towards the BSc and that, to the best of my knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgment has been made in the text.

BRIGHT OPOKU	**10003949**	**…….**	**……**
		Signature	**Date**
COMFORT OSEI	**10004333**	**……**	**…….**
		Signature	**Date**
JEROLD BAIDOO	**10003786**	**……**	**…….**
		Signature	**Date**
BENJAMIN ARHEN	**10002470**	**……**	**…….**
		Signature	**Date**
Certified by:			
DR. KWAME OTENG GYASI		**……**	**……..**
Supervisor		**Signature**	**Date**
Certified by:			
MS. LINDA AMOAKO BANING		…….	…….
Head of Department Name		**Signature**	**Date**

SUPERVISOR'S DECLARATION

I hereby declare that the preparation and presentation of the dissertation was supervised in accordance with the dissertation policy of the Christian service university College.

Dr. Kwame Oteng Gyasi Signed: ……. Date: …….

(Supervisor)

Ms. Linda Amoako Baning Signed:…….. Date: …….

(Head of Department)

ABSTRACT

These days, the introduction of social media applications makes communication simpler and faster. Social media are interactive computer interceded innovations that encourage the creation and sharing of data, thoughts, profession interests and different types of articulation through virtual networks and systems. This project is a social media application based on the Android platform. The application has been named Yɛnkɔnecti. The application is not only used for socialization but can also serve a platform that facilitates monetary transactions and other electronic commerce activities such as buying and selling, a platform that provides users with current news from across the globe and ability to provide users with locations of nearby hospitals, restaurants and fuel stations. The application has a front-end and a back-end. The front end of the Yɛnkɔnecti social media application was developed with Java and XML. In developing the back end of the Yɛnkɔnecti application, the Firebase framework and Google Maps API were used. It is hoped that after this project everyone would be aware that social media application can be used for more than just socialization.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

We give thanks to the Almighty God for his guidance and abundant blessings, for making it possible for us to put this work together in the pursuit of Bachelor of Science degree in Computer Science.

We would also like to express our profound gratitude to our supervisor Dr. Kwame Oteng Gyasi whose encouragement, guidance and support helped us from the initial to the final level and enabled us to develop an understanding of the project.

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Communication has become a key element in terms of building a strong relationship with our family or loved ones. The earliest methods of communicating across great distances used written correspondence delivered by hand from one person to another. [7]

In other words, letters the earliest form of postal service dates back to 550 BC, and this primitive delivery system would become more widespread and streamlined in future centuries. [7] In 1792, the telegraph was invented to allowed messages to be delivered over a long distance far faster than a horse and rider could carry them. Although telegraph messages were short, they were a revolutionary way to convey news and information.

Over the years, many attempts have been made to improve communication amongst ourselves by developing several applications which make communication possible. [7] These days, the introduction of social media applications makes communication much simpler and faster. Social media are interactive computer interceded innovations that encourage the creation and sharing of data, thoughts, profession interests and different types of articulation through virtual networks and systems. [22]

The variety of stand-alone and built-in social media services currently available introduces challenges of definition; however, there are some common features; [22]

- Social media are intuitive Web 2.0 Internet-based applications. [22]

- User-produced content, for example, content posts or remarks, advanced photographs or recordings, and information created through every single online connection, is the backbone of internet based social life. [22]
- Users make administration explicit profiles for the site or application that are structured and kept up by the online networking association. [22]
- Social media encourage the improvement of online informal organizations by associating a client's profile with those of different people or gatherings. [22]

WhatApp is a popular social media application with which more than 1 billion people in over 180 countries use to stay in touch with friends and family, anytime and anywhere. WhatsApp is free and offers simple, secure, reliable messaging and calling, available on phones all over the world. [21]

Also there's Venmo which is a mobile payment service owned by PayPal. It allows users to transfer money to others using the service via a mobile phone app, both the sender and receiver have to live in the US. Venmo can also be termed as a social media app because it provides platforms to communicate with friends and also it was created so friends could quickly split bills, whether that is for movies, dinner, rent, tickets.

Yɛnkɔnecti is an online social media application which is not only used for socialization but also provides a platform that facilitates monetary transactions and other electronic commerce activities such as buying and selling, and has a platform that provides users with current news from across the globe.

1.2 Problem Statement

According to developers of WhatsApp, "it started as an alternative to SMS. Now their product supports sending and receiving a variety of media: text, photos, videos, documents, and location, as well as voice calls. [21] Messages and calls are secured with

end-to-end encryption, meaning that no third-party including WhatsApp can read or listen to them. [21]

Behind every product decision is our desire to let people communicate anywhere in the world without barriers." [21] Even though, WhatsApp accomplishes the goals of most social media application project, WhatsApp provides no platform to support money transfer.

Venmo only allows you to pay and request money from your friends. At its core, Venmo provides a social way to pay your friends when you owe them money and don't want to deal with cash. [20] For example:

- Splitting a lunch bill [20]
- Paying your friend half of a cab fare [20]
- Sending your roommate your half of the rent. [20]

Venmo allows users to transfer money to others using the service using a mobile phone app, both the sender and receiver have to live in the US. [20] Therefore, with all these splendid features Venmo has, it is of no use to anyone living outside the US.

The **Yɛnkɔnecti** social media application would provide users with the ability to make mobile money transactions, perform electronic commerce activities such as buying and selling and would also facilitate easy access to news across the globe in addition to accomplishing the goal of most social media application which is facilitating communication.

1.3 Objectives

The long-haul objective of the exploration is to build up an internet based social application that gives substantially more adaptability not at all like different stages as demonstrated. Particularly, our project has the following sub-objectives:

1. Enhance communication.

2. Enable easy monetary transactions:
    - **Yɛnkɔnecti** would be linked to transferwise to aid in transferring money locally and across the globe.
    - To support the monetary transactions Yɛnkɔnecti would have the Venmo API embedded in its package to allow users of Yɛnkɔnecti in the US to make same transactions Venmo users are allowed to.
3. Provide platform for easy access to current news around the globe.
4. Provide a platform for buying and selling by linking users to electronic market places.
5. Able to provide users with directions to local restaurants, fuel stations and hospitals in their vicinity with the touch of a button.

A preliminary literature review shows that existing social media applications are primarily focused on either communication, providing news platform, providing a platform to conduct e-commerce activities such as buying and selling, trading of files and information but none of them facilitates the sending of mobile money between users, provide directions to local restaurants, fuel stations and hospitals in the vicinity of the users and sending money between users.

They usually facilitate one or two of these features but not all. For example, the documentation of the WhatsApp mobile application states that, "WhatsApp started as an alternative to SMS. Our product now supports sending and receiving a variety of media: text, photos, videos, documents, and location, as well as voice calls. [21] Our messages and calls are secured with end-to-end encryption, meaning that no third-party including WhatsApp can read or listen to them. [21]

Behind every product decision is our desire to let people communicate anywhere in the world without barriers". [21] Clearly, monetary transactions and directional guidance are not part of their agenda.

According to Facebook's corporate vision statement, based on the company's social media service offerings, is "People use Facebook to stay connected with friends and family, to discover what's going on in the world, and to share and express what matters to them." [16] This corporate vision shows that the business is not just a basic online social networking site. Instead, the company also facilitates knowledge acquisition and valuable communication among individual users. [16] Facebook's vision statement has the following main components:

- Global market scope [16]
- Tool for communication [16]
- Tool for discovery [16]
- Tool for self-expression [16]

To the best of our knowledge, Facebook which has at least three of the above features which are to begin with, the main objective which is communication, providing users with the platform to read news online, to advertise products which aids in buying and selling. SnapChat was the only social media app which had almost all of these features in their package, but SnapChat has discontinued its SnapCash feature which allowed monetary transactions.

The SnapCash service was introduced in 2014 – at the peak of SnapChat's popularity – in a partnership with Square as a way to take on payment platform Venmo. [19] [15] Users were able to link a bank account or debit card to their SnapChat profiles and could send or request money in the app's chat feature by writing the desired amount

beginning with a dollar sign. Unfortunately, SnapCash disappeared on 30th August, 2018. [19] [15]

1.4 Methodology

The development of the Yɛnkɔnecti application will entail operational findings. Yɛnkɔnecti will come up with necessary requirements such as system and relational objectives. The actualities and discoveries obtained will help in the complete improvement of the application. The development of the application will involve system analysis, system design, system construction and engineering, system testing and implementation. In other words, in the development of the Yɛnkɔnecti the waterfall model will be used.

The front end of the Yɛnkɔnecti social media application will be developed with:

- XML which Extensible Markup Language will be used to plan the interface of the application.
- Java because it is possible to use Java standard library in addition to the Android library when coding.

The back end of the Yɛnkɔnecti application would be developed with the following:

- Firebase Cloud Functions with JSON (on the firebase server which allows the application to send notifications from one phone to another.
- Firebase Authentication would be used for user sign up and sign in authentications.
- Firebase Realtime Database would be used as a database server.
- Firebase Firestore Database would be also used as a database server.
- Google Maps API would be used for live locations.

- Google Places API would be used to identify places and landmark locations.

CHAPTER 2

LITERATURE REVIEW

2.1 What Is Social Media?

Social media is an expression that we toss around a ton nowadays, regularly to portray what we post on locales and applications like Facebook, Twitter, Instagram, SnapChat and others. In any case, in the event that we utilize the term to portray a site like Facebook, and furthermore a site like Digg, in addition to a site like Wikipedia then it begins to get all the more confounding. Exactly what precisely is online life, at any rate? [10]

The term is utilized so ambiguously that it can essentially be utilized to depict practically any site on the web today. Or on the other hand can it? A few people have all the more a confined perspective via web-based networking media, regularly comparing it to mean equivalent to long range interpersonal communication (otherwise called Facebook, Twitter, and so on.). Other individuals don't think about web journals to fall under the web-based social networking class. [10]

It seems as if everyone has their own personal opinion of what social media is and isn't. But let's dive deeper into the general concept to gain a clearer and more precise understanding. [10]

2.2 All in all, What Is Social Media?

As opposed to characterize the term utilizing a lot of exhausting language that would most likely just muddle things further, maybe the most ideal approach to get a clearer comprehension of it is to separate it into less difficult terms. To begin, how about we take a gander at each word exclusively.

The '**social**' part: alludes to getting together with different people by sharing data utilizing them and accepting data from them. [10]

The '**media**' part: alludes to a guitar of correspondence, similar to the web (while TV, radio, and papers are instances of progressively customary types of media). [10]

From both of these different terms, we can pull a basic definition together: Social media systems administration are electronic specialized apparatuses that empower individuals to cooperate together by both sharing and devouring data. Truly, it's a wide definition — however recollect that online networking is extremely a wide term. This truly is likely as explicit as we can get without focusing in a great deal on a progressively explicit subcategory of social.

2.3 Basic Social Media Features

The accompanying arrangement of basic highlights will in general be dead giveaways of a web based social life web page. In case you're addressing whether a particular site could be named social or not, take a stab at searching for at any rate one of these highlights: [10]

- **Personal client accounts:** If a site enables guests to make their specific records that they can sign into, at that point that is a decent first sign it may be utilized for some sort of client-based communication — maybe social cooperation. Despite the fact that it's conceivable to decently impart data or converse with others online namelessly, making some sort of client account initially is to a greater extent a standard, standard thing. [10]
- **Profile pages:** Since internet-based life is centered around correspondence, a profile page is much of the time important to incredibly help speak to an individual and give them a space to make their specific individual brand. It regularly incorporates insights regarding the individual client, for example

a profile photograph, bio, site, feed of late posts, suggestions, late action and the sky is the limit from there. [10]

- **Friends, supporters, gatherings, hashtags and so forth:** Individuals utilize their records for associating with different clients. They can likewise go through them to sign for specific types of data. [10]
- **News channels:** When clients relate exclusively to different clients via web-based networking media, they're essentially saying, "I might want to get data from these individuals." That data is refreshed for them continuously by means of their news channel. [10]
- **Personalization:** Social media showcasing destinations as a rule give clients the flexibleness to design their client settings, tweak their profiles in a specific style, arrange their companions or devotees, deal with the data they see inside their news sources and even give input about what they do or don't want to see. [10]
- **Notifications:** Any site or application that advises clients about explicit data is unquestionably playing the web-based life game. Clients have absolute command over these warnings and can choose for the sorts of notices they need. [10]
- **Information refreshing, sparing or posting:** If a site or a product gives you a chance to post completely anything, with or without a purchaser account, at that point it's social! It could be a basic content-based message, a photograph transfer, a YouTube video, an associate with articles or whatever else. [10]
- **Like catches and remark areas:** Two of the most widely recognized ways we cooperate via web-based networking media are through catches that

speak to a 'like' in addition to remark segments where we can share our musings. [10]

- **Review, rating or casting a ballot framework:** Besides preferring and remarking, loads of social media locales and applications rely upon the aggregate exertion of town to survey, rate and vote on data they find out about or have utilized. Think about your picked shopping destinations or film audit locales that utilization this web based life include. [10]

2.4 What's the Difference Between Social Media and Social Networking?

As expressed before, heaps of individuals utilize the terms informal communication and long range informal communication reciprocally as though they mean the very same thing. Despite the fact that the thing that matters is unpretentious, they're not the equivalent. Person to person communication is frequently a subcategory of web based life.

The easiest method to know the distinction between long range interpersonal communication and person to person communication is by thinking about the terms media and systems administration independently. Media distinguishes the data you're really sharing — regardless of whether it's a URL to an article, a film, an enlivened GIF, a PDF record, a simple notice or whatever else. [10]

Systems administration, then again, has in regards to who your group of spectators is and the connections you have with them. Your system can incorporate individuals like companions, relatives, partners, anybody from your very own past, current clients, tutors and even total outsiders. [10]

They surely cover, which is the reason it gets befuddling. For instance, you can impart media to your informal organization to assemble likes and remarks — a type of long

range interpersonal communication. In any case, you can even just up vote a connection on Reddit, which is extremely a long range informal communication stage, to significantly assist the city and give your state in the circumstance with no expectation of structure associations with different clients. Still confounded? Endeavor to consider informal communication like natural product. Apples, bananas, oranges, grapes, berries, melons and pineapples are all territory of the more extensive natural product class the very same way that interpersonal interaction, social news, social bookmarking, wikis, sites and private web informing are zone of the more extensive long range interpersonal communication classification. [10]

2.5 Are Traditional Media Also Social Media?

Customary media was referenced before on in this data just to demonstrate more extensive examples of media, yet don't be tricked into feeling that TV, radio, and papers are a piece of web-based social networking. In any event not actually yet completely. The line drawn between the two is gradually diminishing as each keeps on developing.

Long range interpersonal communication doesn't simply give you data yet collaborates with you while giving you that data. This communication is frequently as straightforward as mentioning your remarks or enabling you to cast a ballot on a review, or it is regularly as perplexing as Flixster prescribing motion pictures for you based on the evaluations of other individuals with comparable interests. [10]

Consider ordinary media as a single direction road where you could peruse a paper or check out a composed report on TV, yet you have extremely constrained capacity to give your thoughts on the issue. Person to person communication, on another hand, is extremely a two-way road that gives you the capacity to convey as well. [10]

2.6 Are Blogs a Part of Social Media?

Duplicate blogger distributed an intriguing article quite a while prior, making the contention that web journals are to be sure online networking, despite the fact that that people will in general place them in a classification all autonomously nowadays. In reality, web journals are among the most seasoned sorts of web based life that commanded the web quite a while before we were friending and following everybody on interpersonal organizations.

The principle component highlights which make websites area of internet based life are their client accounts, remark segments, and blog systems. Tumblr, Medium, WordPress, and Blogger are only two or three instances of enormous blog stages that have extremely dynamic network blog systems.

2.7 What Are Some of the Known Issues with Social Media?

Social media isn't all just fun and games with your pals, celebrities you admire, and brands you follow. There are lots of common problems that a lot of major social media platforms haven't totally solved, despite their effort to accomplish so.

- **Spam**: Social media makes it easy for spammers — both real people and bots — to bombard others with content. If you have a Twitter account, you've probably experienced a few spam bot follows or interactions. Likewise, in the event that you run a WordPress blog, you could have gotten a spam comment or two caught by your spam filter. [10]
- **Cyberbullying/Cyberstalking**: Children and teenagers are especially susceptible to cyberbullying since they take more risks in regards to posting on social media. And now that people all interact on social media via our cellular devices, most major platforms ensure it is possible to share our locations, opening the doors for cyber stalkers to a target us. [10]

- **Self-image manipulation**: What an individual post about themselves on social media only represents a tiny portion of these life. While followers may see someone who's happy and living it down via their posts on social media in such a way that makes them feel boring or inadequate in comparison, the reality is that users have the energy to completely control what parts they do and don't desire to broadcast on social media to control their particular self-image. [10]
- **Information overload**: It's not unusual to have over 200 Facebook friends or follow over 1,000 Twitter accounts. With so many accounts to follow and so lots of people posting new content, it's extremely difficult to keep up. [10]
- **Fake news**: Fake news websites promote links to their own totally false news stories on social media to be able to drive traffic to them. Many users do not know that they're fake in the initial place. [10]
- **Privacy/Security:** Many social media platforms still get hacked from time to time despite having good security measures in place. Some also don't offer all the privacy options that users need to keep their information as private as they need them to be. [8]

2.8 What is Android?

Android is the name of the mobile operating-system created by American company Google. It was established on a changed adaptation of the Linux portion and other open source programming. It was made basically for touchscreen cell gadgets, for example, cell phones and tablets. Moreover, Google has additionally created Android TV for TVs, Android Auto for autos, and Wear OS for wrist watches, each with a particular

UI. Variations of Android may likewise be utilized on game consoles, advanced camera models, PCs and different hardware. [2] [6] [12] [17]

It most normally comes introduced on various Smartphone and tablets from a lot of makers offering clients access to Google's own administrations like Search, YouTube, Maps, Gmail and then some. This infers you can without much of a stretch attempt to discover data on the web, watch recordings, scan for headings and compose messages on your telephone, similarly you would on your work area, yet there's something else entirely to Android than these straightforward precedents. Working Systems permit us a ton in most recent 15 years. Starting with high contrast telephones to late PDAs or smaller than usual PCs, portable OS has overcome much away. Especially for advanced cells, Mobile OS has enormously developed from Palm OS in 1996 to Windows pocket PC in 2000 at that point to Blackberry OS and Android. A standout amongst the most broadly utilized portable OS today is Android. [2] [6] [12] [17]

Android is a product pack including not only working framework but rather additionally center product and key applications. Android Inc. was established in Palo Alto of California, U.S. by Andy Rubin, Rich excavator, Nick burns and Chris White in 2003. [14] [5] [3] [8] [11] [13] [18]

At first created by Android Inc., which Google purchased in 2005, Android was uncovered in 2007, with the underlying business Android gadget propelled in September 2008. The working framework has since experienced numerous significant discharges, with the present rendition being 9 "Pie", discharged in August 2018. The center Android source code is perceived as Android Open Source Project (AOSP), and is basically authorized underneath the Apache License. [14] [5] [3] [8] [11] [13] [18]

Android can likewise be related with a suite of restrictive programming created by Google, called Google Mobile Services (GMS) that in all respects often comes pre-introduced in gadgets, which will incorporate the Google Chrome internet browser and Google Search and dependably incorporates center applications for administrations, for example, Gmail, alongside the applying store and advanced dissemination stage Google Play, and related improvement stage. These applications are authorized by makers of Android gadgets affirmed under measures forced by Google, yet AOSP has been utilized as the foundation of contending Android biological systems, for example, Amazon's Fire OS, which utilize their specific reciprocals to GMS. Android has been the smash hit OS worldwide on cell phones since 2011 and on tablets since 2013. As of May 2017, it's more than two billion month to month dynamic clients, the best introduced base of any working framework, and by December 2018, the Google Play store includes over 2.6 million applications.

2.9 HISTORY OF ANDROID

2.9.1 From Android 1.0 to Android 9.0, here's the manner by which Google's OS advanced more than ten years

It wasn't until 2005 that Google obtained Android, Inc., and keeping in mind that very little about Android was known at the time, many accepting it as a sign that Google would use the stage to enter the phone business. In the end, Google entered the cell phone business — however not as an equipment maker. [14] [5] [3] [8] [11] [13] [18]

Rather, it showcased Android to different producers, first grabbing the eye of HTC, who utilized the stage for the absolute first Android telephone, the HTC Dream, in 2008. Starting with this underlying rendition of the working framework running on the HTC Dream, how about we examine how Android has changed beforehand decade. [14] [5] [3] [8] [11] [13] [18]

2.9.2 Android 1.0 — Android Market, Widgets, and Notifications (2008)

The Android period formally started on October 22nd, 2008, when the T-Mobile G1 propelled in the United States. [14] [5] [3] [8] [11] [13] [18] At first, numerous highlights that we couldn't live without today were missing — an on-screen console, multi-contact ability, and paid applications, for example — yet the establishment was set up, and a couple of enduring trademarks of the stage appeared on those absolute first G1s to move off the mechanical production system. [2] [11][13] [18]

Android 1.0 was clearly far less created than the working framework we know and love today, however there are a couple of similitudes. For instance, most concurred that Android basically nailed how to manage notices, and it incorporated the draw down warning window that destroyed the notice framework in iOS. [2] [11][13] [18]

Another notable development in Android is the Google Play Store, which, at the time, was known as the Market. While Apple beat it to the punch by propelling the App Store on the iPhone a couple of months sooner, the truth of the matter is that together they kick-began the possibility of a unified spot to get all your applications — something that is difficult to envision not having now. [2] [11][13] [18]

Aside from the Market, Android 1.0 additionally flaunted the capacity to utilize home screen gadgets, an element that iOS did not have. Truth be told, iOS still doesn't give you a chance to put gadgets on your home screen. Sadly, engineers couldn't make their very own gadgets at the time. [14] [5] [3] [8] [11] [13] [18] That changed in later forms. To wrap things up, the main form of Android had profound coordination with Gmail, an administration that had effectively taken off at the time. [14] [5] [3] [8] [11] [13] [18] The Android time formally started on October 22nd, 2008, when the T-Mobile G1 propelled in the United States. At first, numerous highlights that we couldn't live without today were missing — an on-screen console, multi-contact ability, and paid applications, for instance — yet the structure squares were set up, and a couple of

enduring trademarks of the stage appeared on those absolute first G1s to move off the sequential construction system. [14] [5] [3] [8] [11] [13] [18]

Android 1.0 was clearly far less created contrasted with working framework we as a whole know and love today, however there are dependably a couple of similitudes. For instance, most concurred that Android practically nailed how to adapt to notices, and it incorporated the draw down notice window that blew the notice framework in iOS from the water. [2] [11][13] [18]

Another earth shattering development in Android is the Google Play Store, which, at the time, was known as the Market. While Apple beat it to the punch by propelling the App Store on the iPhone a couple of months sooner, actually together they kick-began the thought of an incorporated spot to get your whole applications — something that is difficult to envision lacking at this point. [2] [11][13] [18]

Notwithstanding the Market, Android 1.0 additionally flaunted the ability to utilize home screen gadgets, a capacity that iOS didn't have. Indeed, iOS still doesn't give you a chance to include gadgets your home screen. [14] [5] [3] [8] [11] [13] [18] Lamentably, designers couldn't make their own special gadgets at the time. That changed in later forms. To wrap things up, the absolute first form of Android had profound reconciliation with Gmail, a site that had effectively expelled at the time. [2] [11][13] [18].

2.9.3 Android 1.5 Cupcake — Third-Party Widgets, On-Screen Keyboard, And A Sugary Name (2009)

Android 1.5 — maybe better known by its codename "Cupcake" — denoted a ton all the more an achievement. [14] [5] [3] [8] [11] [13] [18] It wasn't pretty much the reality it included a few long awaited highlights that were basic to keeping the stage focused. It had been likewise the absolute first form to utilize Google's "sweet" naming show: each significant discharge since Cupcake has been named in the wake of having a sugary dessert in order request. The pattern has proceeded and is relied upon to go on.

One year from now's adaptation 10 Q is characterized to be Android's greatest naming test yet. [2] [11][13] [18]

From multiple points of view, Cupcake was about refinement, cleaning some harsh edges on the UI that had initially propelled. A few of those progressions were about indistinct on the off chance that you weren't attempting to discover them. For instance, the run of the mill Google seek gadget — a staple on numerous users' home screens — picked up a hint of straightforwardness, and the application cabinet was brightened with a fragile weave design under the symbols. [2] [11][13] [18]

Drift over the picture underneath to obtain a feeling of exactly how unobtrusive these progressions were. On the off chance that you utilized a gadget running 1.1 and 1.5 in progression, you may not see anything; in all actuality, however, from content arrangement to shading on the status bar go underneath the blade. [2] [11][13] [18]

Most G1 clients presumably flew past those UI changes without seeing them since the broad rundown of new highlights Google had tossed in was significantly more energizing, discernible, and quickly important in everyday use: [2] [11][13] [18]

An on-screen console by and large, it's astonishing to trust Google could've delivered Android with no kind of delicate console, however that is actually what it did. It clarifies why the absolute first Android gadget at retail was a scene QWERTY slider, and what's more, it clarifies why it wasn't until Cupcake was discharged (in April 2009, some a large portion of a year following the G1 sent) that we saw the absolute first touchscreen-just telephone accessible available, the HTC Magic. [2] [11][13] [18]

Together with the delicate console support, Google made a strong stride: it coordinated the snares basic for outsider designers to create their own one of a kind substitution consoles, which truly is an ability that separated Android from contending stages for a considerable length of time. During the season of Cupcake's discharge, the state

Android delicate console was considered by numerous individuals to slack after iOS for precision and speed, which at last drove OEMs like HTC to rapidly create substitutions without anyone else's input gadgets. Without a doubt, it had been one of numerous first kinds of "cleaning" Android would see. [2] [11][13] [18]

Extensible gadgets. While Android 1.0 and 1.1 in fact included gadgets, their maximum capacity still couldn't seem to be acknowledged in light of the fact that Google hadn't presented the SDK to designers. The main gadgets you had accessible were the few incorporated into the container. That changed in 1.5, and today, many (if not most) of the outsider applications on the stage deliver with various gadgets offered to the client. It is a major ordeal for Android, which keeps on taking joy from unquestionably the most adaptable, extensible home screen of any versatile stage — and that title follows its foundations to the expansion of the element in Cupcake. [2] [11][13] [18]

Clipboard enhancements. Android had an incredibly harsh street to increasing "full" support for copy and paste. The stage in fact bolstered it from the very first moment, however it had been to a great extent constrained by content fields and connections. That implied that content couldn't be replicated out of program windows or Gmail, two spots where you're all around prone to need to do it. Despite the fact that full clipboard ability wouldn't touch base at Gmail for every more form, Cupcake added backing to the program, giving you a chance to duplicate plain content out of a page. [2] [11][13] [18]

Video catch and playback. It's hard to envision a cell phone shipping without the help for shooting video presently, yet that is the circumstance that T-Mobile G1 purchasers initially ended up in. Cupcake would fix the issue, however like Android's worked in delicate console, the working framework's worked in camera interface wound up one of a lot more upbraided components of the stage. It is likewise part that OEMs

immediately supplanted with their own one of a kind improved interfaces, much of the time including support for extra scenes, modes, alternatives, and comforts like touch to center. [14] [5] [3] [8] [11] [13] [18] Furthermore, significantly more. Various updates included clump activities in Gmail (you couldn't erase or file numerous messages without a moment's delay before 1.5), transfer support for YouTube and Picasa, and omnipresent use of contacts' Google Talk status all through the stage in spots like the Contacts screen, the Messaging application, and Gmail. (As it were, this element — synchronization of rich contact data over various applications and screens — would prognosticate the course that Android was moving, especially in 2.0.) [2] [11][13] [18]

2.9.4 Android variant 1.6: Donut

Android 1.6, Donut, folded into the whole world in the fall of 2009. Doughnut filled in a couple of significant gaps in Android's middle, including the ability for the OS to work. [2] [11][13] [18]

Eclair was the absolute first gadget to highlight Google Maps route for instance, commencing what before long turned into the demise of the in-vehicle GPS unit. [14] [5] [3] [8] [11] [13] [18] While Maps has changed a great deal as far back as at that point, a couple of significant highlights turned up in the administration which can be as yet present today, for example, turn-by-turn route and voice direction. There have been turn-by-turn route applications back then, however they unquestionably were costly, which means Google's continue to offer Maps for nothing out of pocket was entirely problematic. [2] [11][13] [18]

The web program in Android Eclair additionally got patched up for the fresh out of the box new working framework. Google added HTML5 backing to the program and the ability to play recordings, putting Eclair comparable to the best versatile web machine at the time — the iPhone. [14] [5] [3] [8] [11] [13] [18] To wrap things up was the lock screen, which got a significant invigorate and enabled clients to swipe to open — equivalent

to on the iPhone. From the lock screen, clients could likewise change the telephone's quiet mode. [14] [5] [3] [8] [11] [13] [18] Android 2.1 chiefly filled in as a nuisance fix discharge which incorporated a changed API, so it shared the Eclair codename. [11] [13] [18] [4]

2.9.5 Android 2.2 Froyo (2010)

Android Froyo was discharged in 2010, and demonstrated why it was a reward to truly have a Nexus telephone. The Nexus One, which was the underlying Nexus telephone to be discharged, was likewise the underlying telephone to have the Android Froyo update. Froyo was pointed more at refining the Android experience, offering clients five home screen boards rather than three, and flaunting an overhauled Gallery application. [11] [13] [18] [4]

There were, be that as it may, a few in the engine upgrades. For instance, Froyo was the underlying form of Android to make portable problem area support. Clients likewise at last got the PIN lock screen, which was ideal for people who didn't generally like the example lock screen which was recently offered in Android. [11] [13] [18] [4]

Android 2.2 acquired some exceptionally looked for after highlights which left loads of Android fans. Not simply was this discharge utilizing another bit, in any case the codebase offered numerous presentation upgrades, the expansion of the JIT compiler for much more speed, support for message pop-ups, better Exchange support, coordination of the V8 JavaScript motor in the program, support for screens of up to 320 ppi, USB tying and Wi-Fi hotspot, voice dialing over Bluetooth and Adobe Flash support. [11] [13] [18] [4]

2.9.6 Android 2.3 Gingerbread (2010)

The Nexus program was at long last entering a one of a kind, and the arrival of Gingerbread affirmed that. Google discover the Samsung-assembled Nexus S for this one, nonetheless, a phone which was delivered from Samsung's profoundly fruitful

Galaxy S. Gingerbread was another huge Android refinement, and it saw an overhaul of Android's stock gadgets and home screen. [11] [13] [18] [4]

Gingerbread additionally accompanied an improved console, which offered new shading for the keys, notwithstanding improved multi-contact support, which enabled clients to press various keys to get into a second console. Last in spite of the fact that not least is that Gingerbread included help for the forward-looking camera — what might us selfie-sweethearts manage without that? [11] [13] [18] [4]

2.9.7 Android 3.0 Honeycomb (2011)

Google had been making waves in the cell phone industry for certain years now, which made Honeycomb a very fascinating discharge absolutely in light of the fact that it was outfitted towards tablets. It had been even first displayed on a Motorola gadget that would in the end become the Xoom. [11] [13] [18] [4]

Honeycomb gave a few structure prompts with respect to what appears in future adaptations of Android. As opposed to complementing the os with the exemplary green Android shading, for instance, Google changed to blue accents. In addition, rather than encountering to pick home screen gadgets from a simple rundown, where you couldn't perceive what the gadgets resembled, sneak peaks were offered for individual gadgets. Maybe the greatest move around in Honeycomb was the way that it evacuated the necessity for the physical catch. Rather, the house, back, and menu catches were altogether incorporated into the product as virtual catches, which means they are regularly covered up or appeared on the application. [11] [13] [18] [4]

2.9.8 Android 4.0 Ice Cream Sandwich (2011)

The Nexus S was an incredible telephone, anyway it wasn't the most important thing in the world Google's association with Samsung. The two combined up once more for the arrival of the Galaxy Nexus, which exhibited Ice Cream Sandwich, an os that brought a great deal of Honeycomb's highlights over to the cell phone. For instance,

the operating system brought over the previously mentioned virtual catches, notwithstanding the changed and refined interface that utilized the blue features. Other little highlights, such as face open, information use investigation, and new applications for mail and schedule, were additionally incorporated into the update. [11] [13] [18] [4]

2.9.9 Android 4.1 Jelly Bean (2012)

Android Jelly Bean flagged a spic and span period for the OS, despite the fact that the OS appeared to be just about simply like its ancestor. On the off chance that you burrowed just somewhat more profound; you'd have seen some fundamental changes. The most significant of which was Google Now, which could be gotten to with a quick swipe from the house screen and brought data — for example timetable occasions, messages, and climate projections— all to a solitary screen. The element truly was Google's first real cut at a computerized right hand, and it laid the foundation for future adaptations of advanced colleagues, including Google Assistant. [11] [13] [18] [4] Other than Google Now, various other significant increments were actualized in Jelly Bean, such as Project Butter, which was gone for radically improving Android's touch execution by significantly increasing buffering illustrations. This disposed of a great deal of the falter in Android and made it a much smoother experience generally speaking. Invigorated textual style, expandable notices, more noteworthy gadget adaptability, and different highlights were additionally included Jelly Bean, rendering it among the greatest updates to Android up until this point. [11] [13] [18] [4]

2.9.10 Android 4.4 Kitkat (2013)

The dispatch of Android 4.4 KitKat agreed with the dispatch of the Nexus 5, and it accompanied various extraordinary highlights. For instance, KitKat spoke to among the greatest tasteful changes to the OS to date, modernizing the presence of Android. The blue accents present in Ice Cream Sandwich and Jellybean were supplanted with an unmistakably progressively refined white emphasize, and various the stock

applications that accompanied Android were updated to display lighter shading plans. [11] [13] [18] [4]

Other than a spic and span look, KitKat likewise brought such things as the "alright, Google" seek order, which enabled the buyer to get into Google Now whenever. Moreover, it brought a fresh out of the box new telephone dialer, full-screen applications, and a spic and span Hangouts application, which offered SMS support alongside help for the Hangouts informing stage. [11] [13] [18] [4]

2.9.11 Android 5.0 Lollipop (2014)

Android Lollipop, which appeared close by the Nexus 6, was the first to highlight Google's "Material Design" reasoning. The updates, be that as it may, weren't simply tasteful — the OS additionally displayed a few noteworthy updates underneath the hood. [11] [13] [18] [4]

Google supplanted the maturing Dalvik VM with Android Runtime, for instance, which bragged ahead time arrangement. This basically implied piece of the handling force required for applications could be led before said applications were ever opened. What's more, we saw various warning updates, the expansion of RAW picture support, and various different refinements. Android 5.0 likewise observed the expansion of another rendition of Android, named Android TV, which carried Android to the goliath screen and stays being used on loads of TVs today. [11] [13] [18] [4]

2.9.12 Android 6.0 Marshmallow (2015)

Android Marshmallow achieved both structure changes and changes underneath the hood. For example, the application menu totally changed. Google utilized a splendid foundation rather than dark, for example, and added a hunt bar to help clients rapidly discover the application they required. Android Marshmallow likewise brought the expansion of the memory chief, which enabled you to keep an eye on the memory use of any application utilized inside days passed by 3, 6, 12, or 24 hours. [11] [13] [18] [4]

Next up were the sum controls. In Marshmallow, you gain admittance to an unmistakably increasingly extensive arrangement of volume controls, giving you a chance to change the sum for the gadget, media, and alerts. Security likewise got a serious huge lift inside the working framework. Android formally upheld unique mark sensors you begin with Marshmallow, and consents got a huge patch up. As opposed to applications mentioning all consents forthright when downloaded, authorizations were mentioned on a for each consent premise when these were required. [11] [13] [18] [4]

2.9.13 Android 7.0 Nougat (2016)

Android 7.0 Nougat apparently set apart among the greatest moves up to Android in its 10 years — to a great extent because of how shrewd the OS got. All things considered, likely the greatest change to Android in Nougat is that Google Now was supplanted with the now much-regarded Google Assistant. [14] [5] [3] [8] [11] [13] [18] Close by Assistant, Nougat brought an improved warnings framework, which changed what notifications looked like and acted inside the OS. Notices were displayed from screen to screen, and not at all like past cycles of Android, they could be assembled for brisk administration. Performing multiple tasks likewise got a lift with Nougat. Regardless of whether you're utilizing a telephone or even a tablet, you'll figure out how to utilize split-screen mode, empowering you to utilize two applications in the meantime without exiting out of each application at regular intervals. [14] [5] [3] [8] [11] [13] [18]

2.9.14 Android 8.0 Oreo (2017)

Android Oreo took the Android stage to variant 8.0, and especially brought a huge amount of performing multiple tasks highlights. Picture-in-picture and local split-screen both made their introductions in Android Oreo, which means you may keep watching your favored show on Netflix while perusing the web. [14] [5] [3] [8] [11] [13] [18] Android Oreo additionally gave us significantly more command over notices. With Oreo, clients were enabled to turn warning directs on or off, which means you

approach excessively granular with which notices arrive and what goes on when they show up. Especially, warning channels enabled clients to sort notices predicated on significance. Likewise warning related, Oreo brought notice specks, and the capacity to nap notices. Extra littler highlights turned up in Oreo, as well. For instance, Google got rid of the mass style for emoticons, supplanting them with emoticons which were a slight bit more as per different stages. Oreo likewise gave us auto-empower Wi-Fi, a great content selector, etc. [14] [5] [3] [8] [11] [13] [18]

2.9.15 Android 9.0 Pie (2018)

Presently, ten years following the dispatch of Android on cell phones, we're at Android 9.0 Pie. Android Pie carries with it various visual changes — to such an extent that from an unmistakable point of view, it's the greatest change to Android in two or three years. [14] [5] [3] [8] [11] [13] [18]

Most prominently, Android 9.0 Pie gets rid of the three-catch setup that has existed in Android for a considerable length of time, supplanting it with only one pill-formed catch and signals for controlling things, for example, performing multiple tasks. Android 9.0 Pie likewise carries with it a few changes to warnings, including additional control on the kinds of notices that report up and where they touch base, just as Google's new "Computerized Wellbeing," an element that basically discloses to you how regularly you utilize your telephone, the applications that you utilize the most, etc. The component is focused at helping clients better deal with their advanced lives and control cell phone dependence. [14] [5] [3] [8] [11] [13] [18] Different highlights incorporate versatile battery, which points of confinement just how much battery foundation applications may use, just as "Application Actions," which are profound connects to certain application includes that report up straight from the application cabinet. That is obviously a concise history of Android to date. The portable OS simply

hit its 10-year commemoration, notwithstanding the present form being called Android 9.0 Pie. [14] [5] [3] [8] [11] [13] [18]

2.10 Android App Development

The state language for Android development is Java. Large elements of Android are written in Java and its APIs are created to be called primarily from Java. It's possible to develop C and C++ app using the Android Native Development Kit (NDK), however it isn't something which Google promotes. Based on Google, "the NDK won't benefit most apps. As a developer, you'll need to balance its benefits against its drawbacks. Notably, using native code on Android generally does not cause a noticeable performance improvement, however it always increases your app complexity." [4] [9] [1]

The languages you could consider learning for Android development include:

Java – Java is the state language of Android development and is supported by Android Studio. It features a steep learning curve however. [4] [9] [1] Kotlin – Kotlin was recently introduced as a secondary "official" Java language. It is similar to Java in many ways but is really a little easier to really get your head around. [4] [9] [1] C/C++ — Android Studio also supports C++ with the use of the Java NDK. This allows for native coding applications, which can be handy for things like games. C++ is harder though. [4] [9] [1]

C# — C# is really a slightly more beginner-friendly alternative to C or C++ that obfuscates more code. It's supported by some very handy tools like Unity and Xamarin which are great for game development and for cross-platform development. [4] [9] [1]

BASIC – A plus option is to learn BASIC and try the B4A IDE from Anywhere Software. This is an easy but powerful tool, though definitely a great deal more niche! [4] [9] [1]

Corona/LUA – Another cross-platform tool build on LUA. It massively simplifies the app-building process and lets you call native libraries. [4] [9] [1]

PhoneGap (HTML, CSS, JavaScript) – If you already learn how to build interactive webpages, then you should use this knowledge with PhoneGap to construct a more basic cross-platform app. [4] [9] [1]

2.10.1 Java

In regards time to develop Android apps, the first and hottest option is Java. Java is the state language of Android development, meaning it's the one which has the most support from Google and the one which most apps on the Play Store are built with. [4] [9] [1]

The main way to develop Android apps, is to go ahead and download Android Studio. This is a software program called an IDE, or Integrated Development Environment. It can come packaged with the Android SDK (a pair of tools to facilitate Android development specifically) and basically this will give you everything required in one single place to have up and running. The state tutorials and documentation from Google will reference this technique and you will find the largest number of libraries (free code to improve your personal apps) and tutorials that focus with this method. [4] [9] [1] Java itself was released by Sun Microsystems in 1995 and is used for a wide range of programming applications. Java code is run with a "virtual machine," which runs on Android devices and interprets the code. [4] [9] [1]

Unfortunately, Java can be a little complicated and it's not just a great "first language." It's this that can provide the greatest barrier for many people who wish to get started with Android development, in fact. Android is a thing-oriented programming language with confusing topics like constructors, null pointer exceptions, checked exceptions and more. It's not terribly readable and you'll use a lot of "boiler plate" code doing simple things. Include the Java SDK and things get more complicated still – a first-

time coder can struggle to understand what's Java and what's Android Development using this route also takes a basic knowledge of concepts like Gradle, just like the Android Manifest and the markup language XML. [4] [9] [1]

That's not to imply that Java is really a bad language – definitely not it. Not only wouldn't it be wrong to call any language "bad," but it is also true that a lot of the inconveniences of Java are in fact there for our own good and encourage clean code. A lot of people love Java because of this, and it is also one of the very most versatile and widely used. Based on the PYPL (PopularitY of Programming Languages) table, Java is the most sought after programming language among employers. [4] [9] [1]

Making life a whole lot simpler is Android Studio that has been going from strength to strength over the last few years. Features like a visual designer and suggestions make the process a fair bit smoother, while advanced, powerful features are increasingly being added all the time to offer developers usage of things like cloud storage with easy implementation. It's worth getting aboard, even if this rapid progress does make it hard to maintain sometimes. [4] [9] [1]

2.10.2 Kotlin

Kotlin recently burst onto the scene because the "other" official language for Android development. Some speculation suggests that will probably enhance the language's profile and that it may possibly become the following Swift. [4] [9] [1]

Like Java, Kotlin runs on the Java Virtual Machine. Additionally, it is completely interoperable with Java and doesn't cause any decelerate or escalation in file sizes. The difference is that Kotlin requires less "boiler plate" code, meaning that it is an even more streamlined and easy-to-read system. In addition, it does away with errors like null point exceptions and even excuses you from ending every line with semi colons. Simply speaking, it's great if you're just understanding how to develop Android apps for the first time. [4] [9] [1]

So Kotlin is definitely an easier kick off point for novices, and the fact you are able to still use Android Studio is really a big plus. It's still not quite as simple to pick up as say C# with Unity though, and the community support is in its relative infancy. In fact, you currently have to download a beta version of Android Studio in order to get the out-the-box support. [4] [9] [1]

Still, Kotlin should definitely be on your radar and could offer a less strenuous entry point out “proper” Android Development. That will be likely why Google introduced it in the first place. [4] [9] [1]

2.10.3 C/C++

It's fair to say that most people reading this would not choose this route to develop Android apps. Android Studio offers support for C/C++ code utilizing the Android NDK (Native Development Kit). This means you'll be writing code that doesn't operate on the Java Virtual Machine but rather runs natively on the device and offers you more control over things like memory. For intensive applications like 3D games, this could allow you to squeeze extra performance out of an Android device. It entails you'll have the ability to use libraries written in C or C++. [4] [9] [1]

However, in addition, it is commonly much harder to setup, it introduces more bugs and it's less flexible. And in the event that you did want to make a computer game, you'd of times be better off using a ready-made game engine. [20] [21] [22]

2.10.4 C#

C# is basically a less strenuous, pure object-oriented version of C and C++ developed by Microsoft. It aimed to create the energy of C++ and the easy Visual Basic and reads a little such as a simplified version of Java. Like Java, C# is garbage collected, meaning you never have to be concerned about things like memory leaks and freeing up memory yourself. At the same time though, C# is more modern than Java with a solution syntax

– though this may just be my own bias coming through. The most effective language to develop Android apps often just comes down seriously to taste. [4] [9] [1]

If you will want particularly easy and welcoming introduction to Android app development though, then I would suggest the combination of C# and Unity. Unity is really a “game engine” (meaning it provides things like physics calculations and 3D graphics rendering) and an IDE like Android Studio. This is a free tool that makes it incredibly easy to produce your personal games – with just a few lines of code you could have a fundamental platform game put up in under an hour. No exaggeration. And it's perfectly powerful too, being the tool utilized by most game studios on the Google Play Store. And its multi-platform too. Together with all that, developing in this manner provides an extremely practical way to understand Object Oriented coding (because the objects in this case actually are objects a lot of the time). [4] [9] [1]

The limitation? Unity is useful for creating games but sub-par for creating standard Android Apps, particularly if you wish to conform to Google's Material Design language. If you wish to develop into a professional Android Developer, then this non-standard route will limit your employment opportunities – unless your aim is becoming a game developer, in which case this is a perfectly relevant and professional background. [4] [9] [1]

Not keen on Unity? Then you could consider Unreal instead (better graphics, less suited to mobile) or simplified game-makers like GameMaker Studio. [4] [9] [1]

C# may also be used with Xamarin through Visual Studio. This is more akin to traditional Android development with the advantage of being cross platform (one codebase for Android and iOS). For a complete beginner, this route is again a slightly obtuse entry point to Android development – but for a small company wanting to create

an app for iOS and Android, it makes sense, and there's plenty of support and information out there to help you out. [4] [9] [1]

2.10.5 Basic

Remember how I said that C# was an effort to provide the ability of C with the easy Visual BASIC? Well that's because BASIC (Beginners All-Purpose Symbolic Instruction Code) is incredibly pleasant to use and a totally ideal jumping on point for understanding how to code. [4] [9] [1]

Unfortunately, it isn't officially support by Android Studio and nor are you able to utilize it in Unity or Xamarin. The good news is that there is a lesser-known selection for developing Android apps in BASIC called B4A from Anywhere Software. This really is an acronym for 'BASIC 4 Android' and as you might expect, it enables you to code Android apps with BASIC. It's most certainly not the first selection for most programmers who would like to develop Android apps, but it's always nice to have more options. [4] [9] [1]

B4A is made as a RAD, or Rapid Application Development environment. There are lots of other smart design decisions to make life easier, and there's a very supportive community when you have any questions. [4] [9] [1] This is a superb way to learn to code, and you can build some pretty powerful apps that way alone. It's not well suited for making higher-end games though and once more is suffering from as an "unofficial" option – so it's harder to generate something which exactly meets the Material Design specifications and you'll find it harder working as a specialist developer with only BASIC. One other big drawback is that this is actually the only option on the list that isn't free. [4] [9] [1]

2.10.6 Corona

Corona offers another considerably simpler selection for developing Android apps while still providing you a reasonable number of power and control. You'll be coding in LUA that is already easier than Java and on top of that, the Corona SDK (Software Development Kit) is likely to make things even easier. It supports all native libraries, allows you publish to multiple platforms. It is essentially useful for creating games but can be utilized in many different alternative methods too. You'll need to employ a text editor like Notepad++ to enter your code and you can run said code on an emulator without even having to compile first. When you're ready to generate an APK and deploy, you'll manage to try this utilizing an online tool. [4] [9] [1]

This does require basic coding skills, but it supplies a nice and gentle introduction to the planet of programming. At the same time though, it is definitely limited to some degree and is really a few steps taken from stepping into “app builder” territory. This really is more ideal for someone who wants to create something relatively simple and isn't as concerned about developing their coding skills or learning to be a pro. If you wish to use features such as in-app purchasing, then you will need to pay for a fee. The same goes for using native Android APIs. [4] [9] [1]

2.10.7 PhoneGap

Finally, the last major “simplified” option you can to turn to develop Android apps is PhoneGap, unless you wish to turn instead to a software builder program. PhoneGap is powered by Apache Cordova and essentially lets you create apps utilizing the same code you'd normally use to produce a website: HTML, CSS and JavaScript. This really is then shown via a “WebView” but packaged such as an app. PhoneGap then acts like a connection, allowing developers to get into some basic native features of the device or tablet – such as the accelerometer or the camera. [4] [9] [1]

This isn't really “true” Android development though, and the only real programming will undoubtedly be JavaScript. For all basic tasks, it will get the job done, but when you wish to manage to claim true “Android app developer-hood” (that's a thing), then you definitely should brave one of many other choices with this list. [4] [9] [1]

2.11 Preliminary Research Methodology

The development of the Yɛnkɔnecti application will entail operational findings. Yɛnkɔnecti will come up with necessary requirements such as system and relational objectives. The facts and findings acquired will help in the total development of the application. The development of the application will involve system analysis, system design, system construction and engineering, system testing and implementation. In other words, in the development of the Yɛnkɔnecti the waterfall model will be used.

The front end of the Yɛnkɔnecti social media application will be the developed with:

- XML which Extensible Markup Language will be used to design the interface of the app.
- Java because it is possible to use Java standard library in addition to the Android library when coding.

The back end of the Yɛnkɔnecti application would be developed with the following:

- Firebase Cloud Functions with JSON (on the firebase server which allows the application to send notifications from one phone to another.
- Firebase Authentication would be used for user sign up and sign in authentications.
- Firebase Realtime Database would be used as a database server.
- Firebase Firestore Database would be also used as a database server.
- Google Maps API would be used for live locations.

CHAPTER 3

RESEARCH METHODOLOGIES AND DESIGN

This chapter is a walk-through of how the Yɛnkɔnecti application social media application was developed. The following are some technical terms to be used throughout this chapter;

1. Activity which means a page in android.
2. Fragment is a portion of an activity.
3. Views which are the objects in the activity of fragments. Examples buttons, image views, Linear Layout etc.
4. OnClick listener which is a method which is called when a view is clicked.

The front end of the Yɛnkɔnecti social media application was developed with:

1. XML which is Extensible Markup Language was used to design the interface ("activities and fragments") of the app.
2. Java because it is possible to use Java standard library in addition to the Android library when coding.

The back end of the Yɛnkɔnecti application was developed with the following:

1. Firebase Cloud Functions with JSON (on the firebase server which allows the application to send notifications from one phone to another.
2. Firebase Authentication would be used for user sign up and sign in authentications.
3. Firebase Realtime Database would be used as a database server.
4. Firebase Firestore Database would be also used as a database server.
5. Google Maps API would be used for live locations.

6. Google Places API would be used to identify places and landmark locations.

3.1 Splash Screen Activity

Basically, what this activity does is, it delays the running of application by 5 seconds. This is the launcher activity, meaning it is the first activity to be opened whenever the application is launched or started. From this activity the next activity could either be the main activity or the welcome activity which contains buttons which would also lead to either the sign in or sign up activities. Within the 5 seconds delay the application determines whether there is currently a user signed in or not. If there is a user signed in, the next activity should be the main activity which should be opened up automatically when the 5 seconds is up else the next activity should be the welcome activity should open up after 5 seconds. Both the xml and the java codes for the splash screen activity can be found in "**Appendix A**".

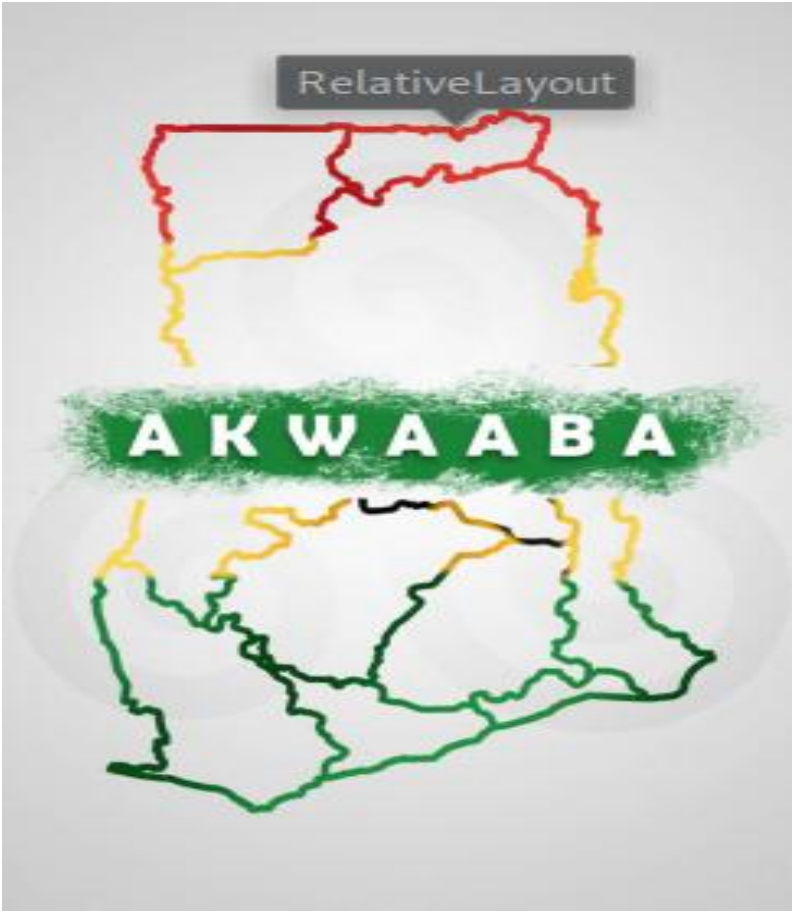

Figure 3. 1 Splash Screen Layout

3.2 Welcome Activity

This activity is the next activity after the splash activity if there is no user currently signed in. This activity has two main buttons which are the “SIGN IN” button and the “SIGN UP” button. When the “SIGN IN” button is clicked, a new activity called the “LoginPageActivity” is opened from there the registered users would be able to sign in which their credentials. When the “SIGN UP” button is clicked, a new activity called the “RegistrationPageActivity” activity is opened from there the unregistered users would be able to sign up which their credentials.

To see the codes for the below activity, refer to “**Appendix A**”.

Figure 3. 2 Welcome Screen layout

3.3 Login Page Activity

This activity provides the user with two editText boxes for their credentials, which is one for their email address and the other for their password. When the boxes with the right information and the “LOGIN” button is clicked, the “Signin” method is called. The “Signin” method when called gets the values from then email field and password field from the user and then passes these values to the Sign in method of Firebase Authentication which verifies the credentials and the signs the user in on approval. Again, when the user clicks the “forgot password textview” a new activity is opened for the reset of the password.

To see the codes for the above activity, refer to “**Appendix A**”.

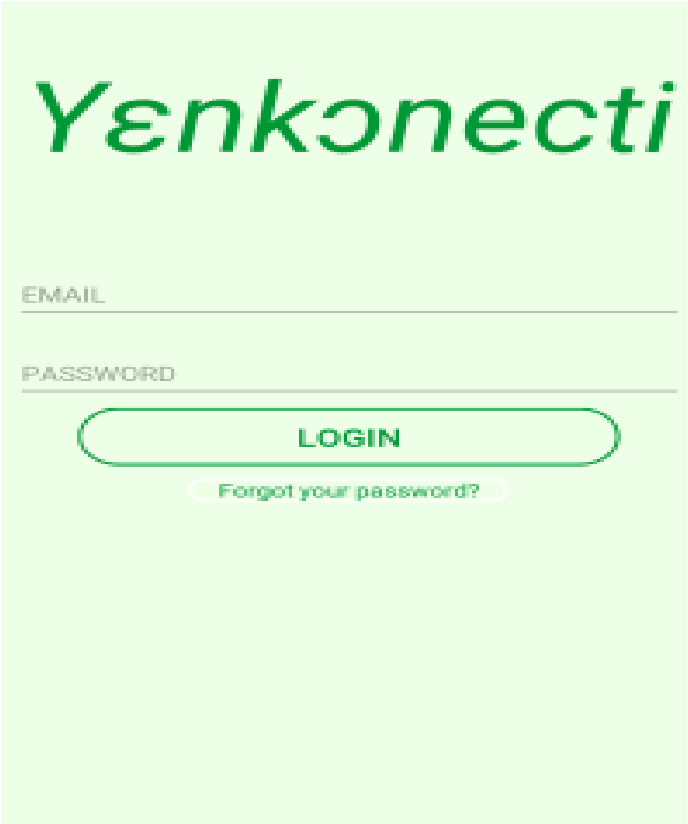

Figure 3. 3 Login Screen Layout

3.4 Registration Page Activity

The above image describes how the "Registration Page Activity" looks. This activity allows the prospective users to create a new account with their credentials. The method "CreateUser" gets the credential the user provides in the editText boxes and registers the user by passing the values to the firebase method sign up after which it stores some of these credentials in the Firebase Database and Firebase Storage. This method is called when the user clicks the "DONE" button. This activity has been designed in such a way that even if one box is not filled, the "CreateUser" would not be called and the user's attention would be drawn to the unfilled box.

To see the codes for the above activity, refer to "**Appendix A**".

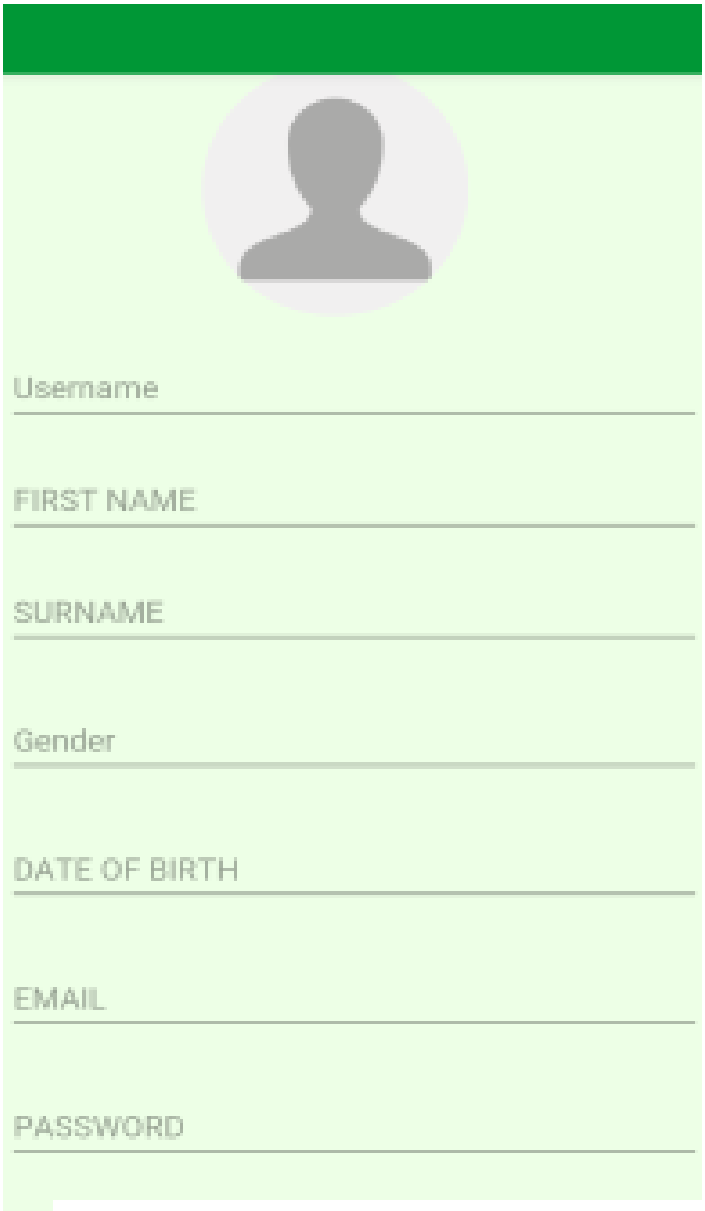

Figure 3. 4 Registration Screen Layout

3.5 Reset Password Activity

This activity allows users to request for a password reset link when they forget their passwords. When the “RESET” button is clicked, a method named the “Reset” is called. This method gets the email address from the user then passes the email address as an argument to the firebase method “sendPasswordResetEmail” which send as a reset password link to the email of the user.

To see the codes for the above activity, refer to “**Appendix A**”.

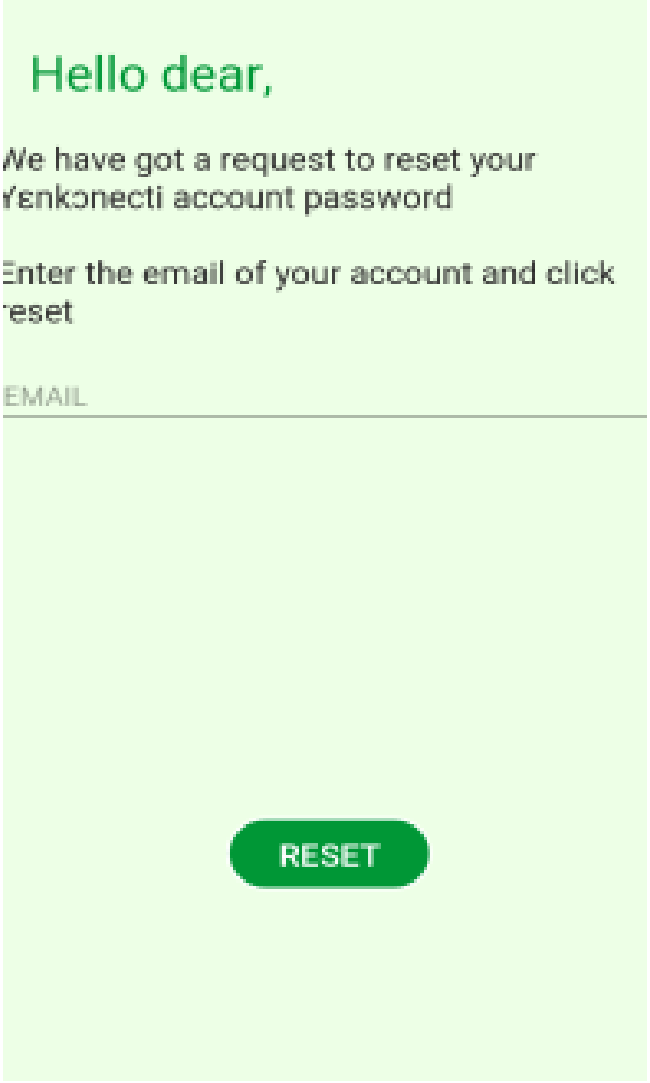

Figure 3. 5 Reset Screen Layout

3.6 Mainactivityy Activity

The figure below describes the design or looks of the activity. The “MainActivityy” is the parent activity of all other activities besides the already mentioned ones. This activity contains a lot of fragments. The user can navigate to almost every other activity and fragment. The activity gets data from the database in the “OnCreate” and “OnStart” methods of the activity.

To see the codes for the above activity, refer to “**Appendix A**”.

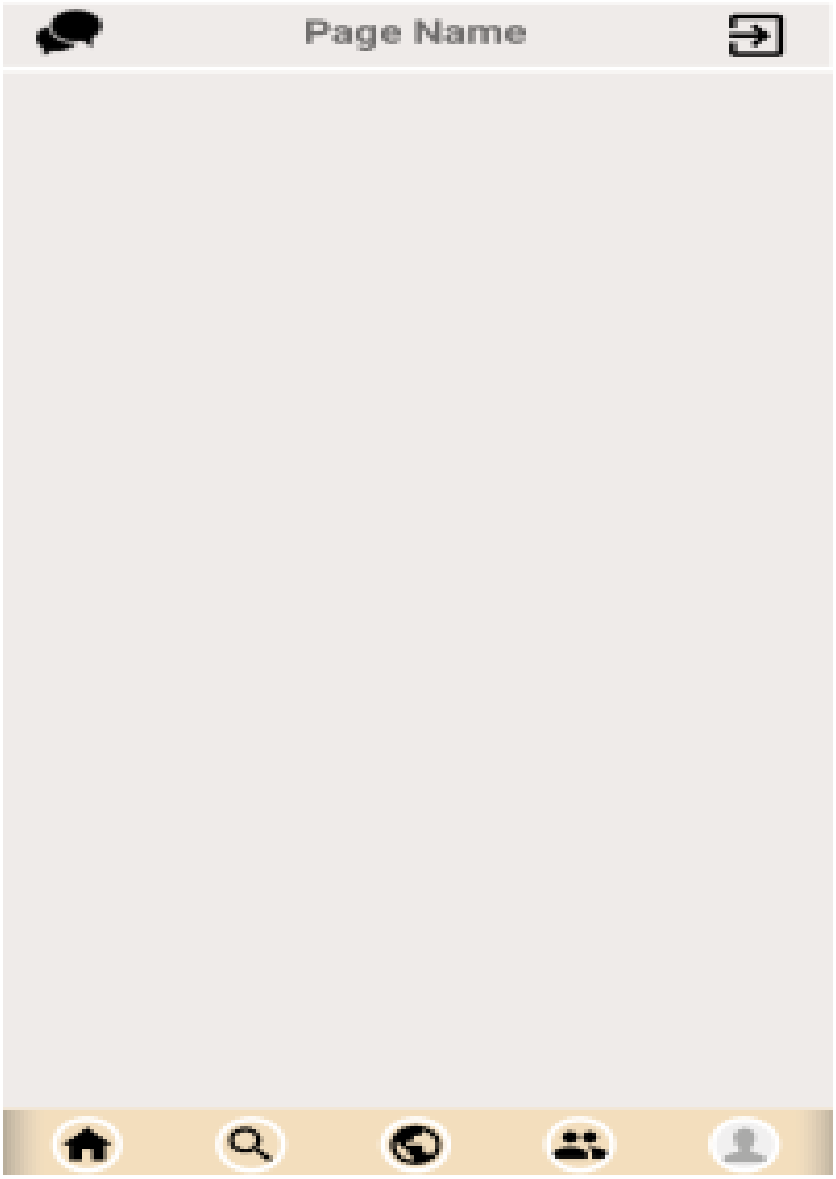

Figure 3. 6 Mainactivityy Screen Layout

3.7 Home Fragment

The home fragment is a fragment contained the “MainActivityy” which contains a view called the recyclerview, an editText box and two buttons known as the “SEND” and “Photo” button. The recyclerview contains the posts from the current user’s friends be it pictures and text messages. The editText box allows users to post only texts and the “Photo” button starts an activity which helps users to share images to their stories.

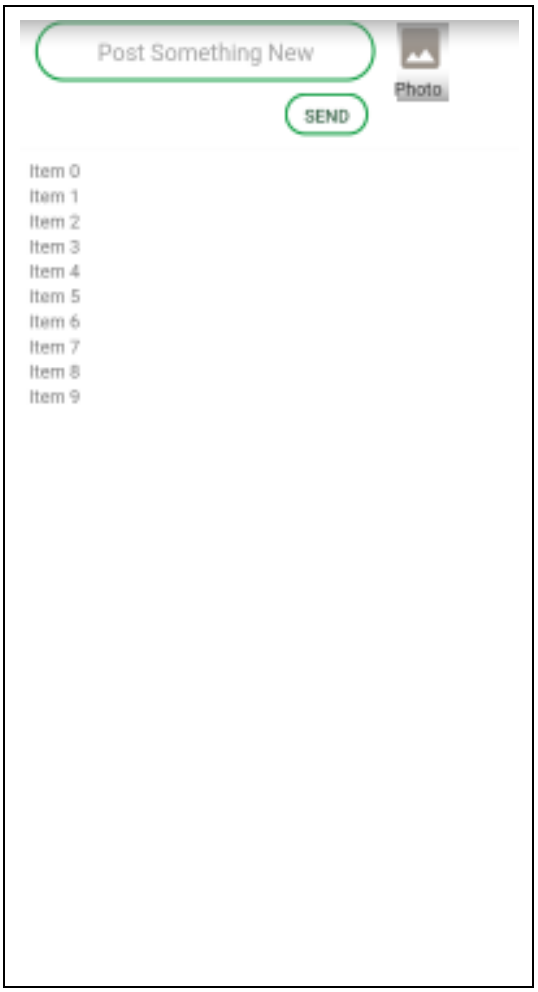

Figure 3. 7 Home Fragment Screen Layout

3.8 Explore Fragment

The explore fragment is another fragment contained the “MainActivityy” which contains a view called the recyclerview and a searchbox. The recyclerview contains the lists of users of the application. Here users are able to look for the profiles of other users and add them as friends either by entering their usernames in the searchbox or by scrolling throw till the find them. This fragment uses a gridview linear recyclerview which means the list of users appear in a linear form not in a grid-like form.

Figure 3. 8 Explore Fragment Screen Layout

3.9 Yɛnkɔnecti World Fragment

The figure below basically describes the layout of the fragment and actions of the views in fragment respectively. The “Monetary Transaction” button opens a new activity from which the transactions money can be done. The “Trade” button opens a new activity in which the e-commerce activity like buying and selling can be done through. The “news” button opens a new activity in which the users gets the latest content from Aljazeera to read. The “locator” button opens the maps activity in which the user can locate nearby fuel station, restaurants and hotels at the touch of a button. The codes in “Appendix A” are XML and JAVA which basically describes the layout of the fragment and actions of the views in fragment respectively.

Figure 3. 9 Yenkonneti World Fragment Screen Layout

3.10 Monetary Transaction Activity

This activity has only two buttons which is the “Transfer Wise” button and the “World Remit”. Both of these buttons when clicked would lead to a money transfer platform which facilitates transfer of funds across the globe. When the “Transfer Wise” button is clicked users would be given the opportunity to register for a transferwise account if they don’t already have, on the other-hand users you already have accounts with transferwise would be able to begin transactions immediately, so as it is with the world remit.

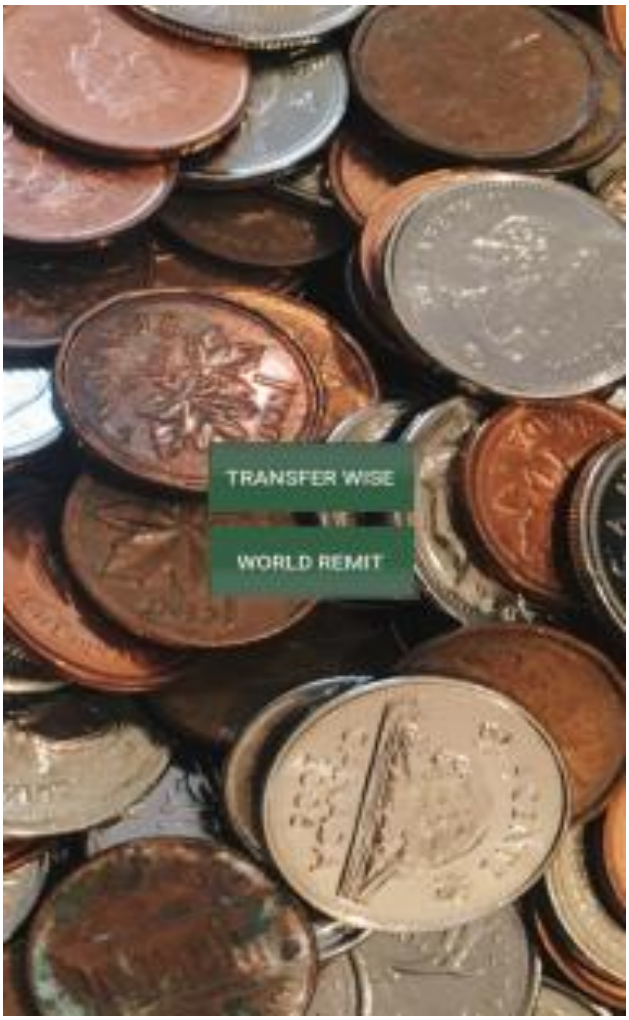

Figure 3. 10 Monetary Transaction Fragment Screen Layout

3.11 Location Activity

This is the maps activity; it is from this activity where users can search for locations they would like to visit. Basically, this activity works just like the google maps just because it uses Google's maps API. Users are able to locate nearby hospitals, fuel stations and restaurants all at the click of a button.

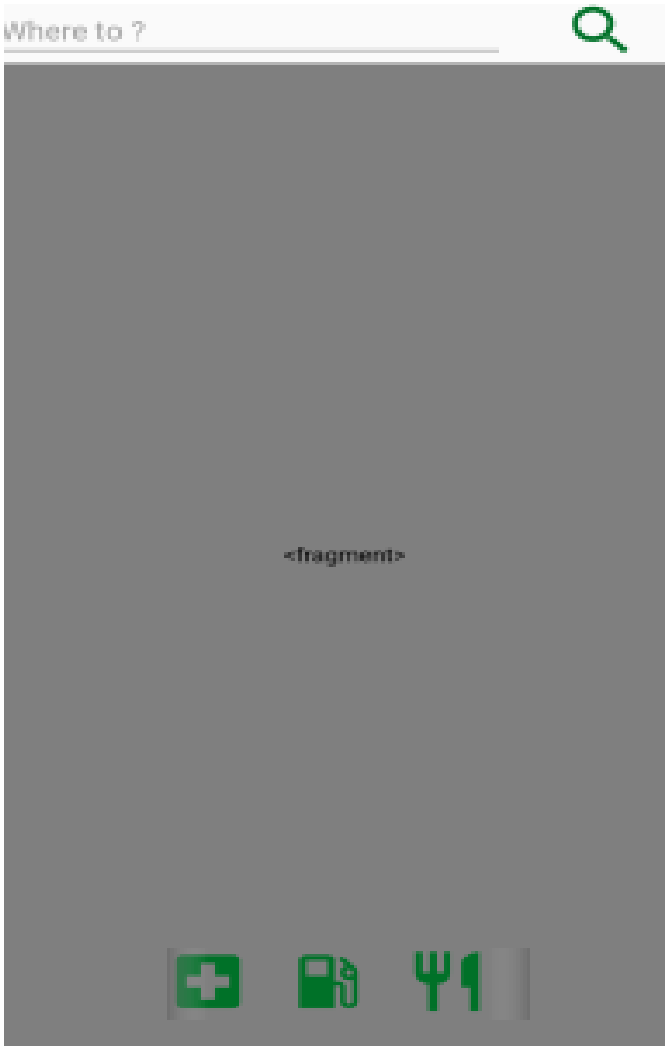

Figure 3. 11 Location Activity Screen Layout

3.12 Friends Fragment

This fragment has a single view which is the recyclerview, the fragment is designed to get the list of ids from a table in the database named “Friends” then get all the details of these ids example, name, picture, status from another table in the database named “Users”. After getting all these details, it binds them onto its recyclerview. Also when the current user receives a new friend request, a button pops in the friends’ fragment indicating that the user has friend requests to respond to.

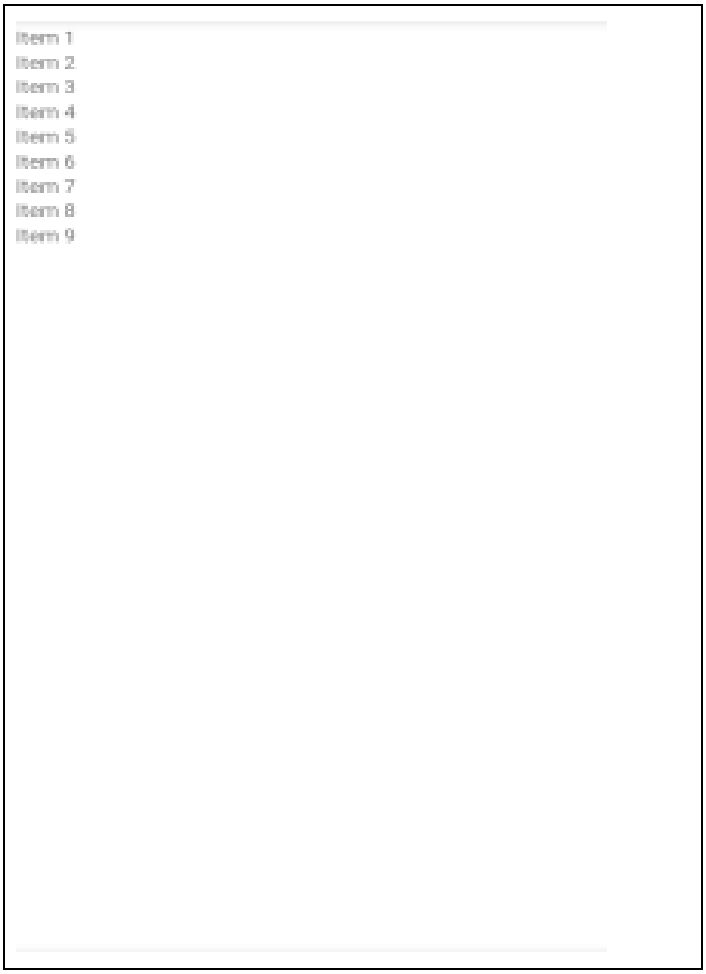

Figure 3. 12 Friends Fragment Screen Layout

3.13 Profile Fragment

Basically, this fragment was designed to get the all the details of the current user from the table named “Users” in the database. The animation of the page starts when the background picture is clicked and ends after the profile fragment has been fully expanded. This fragment has two separate xml design codes for animation purposes.

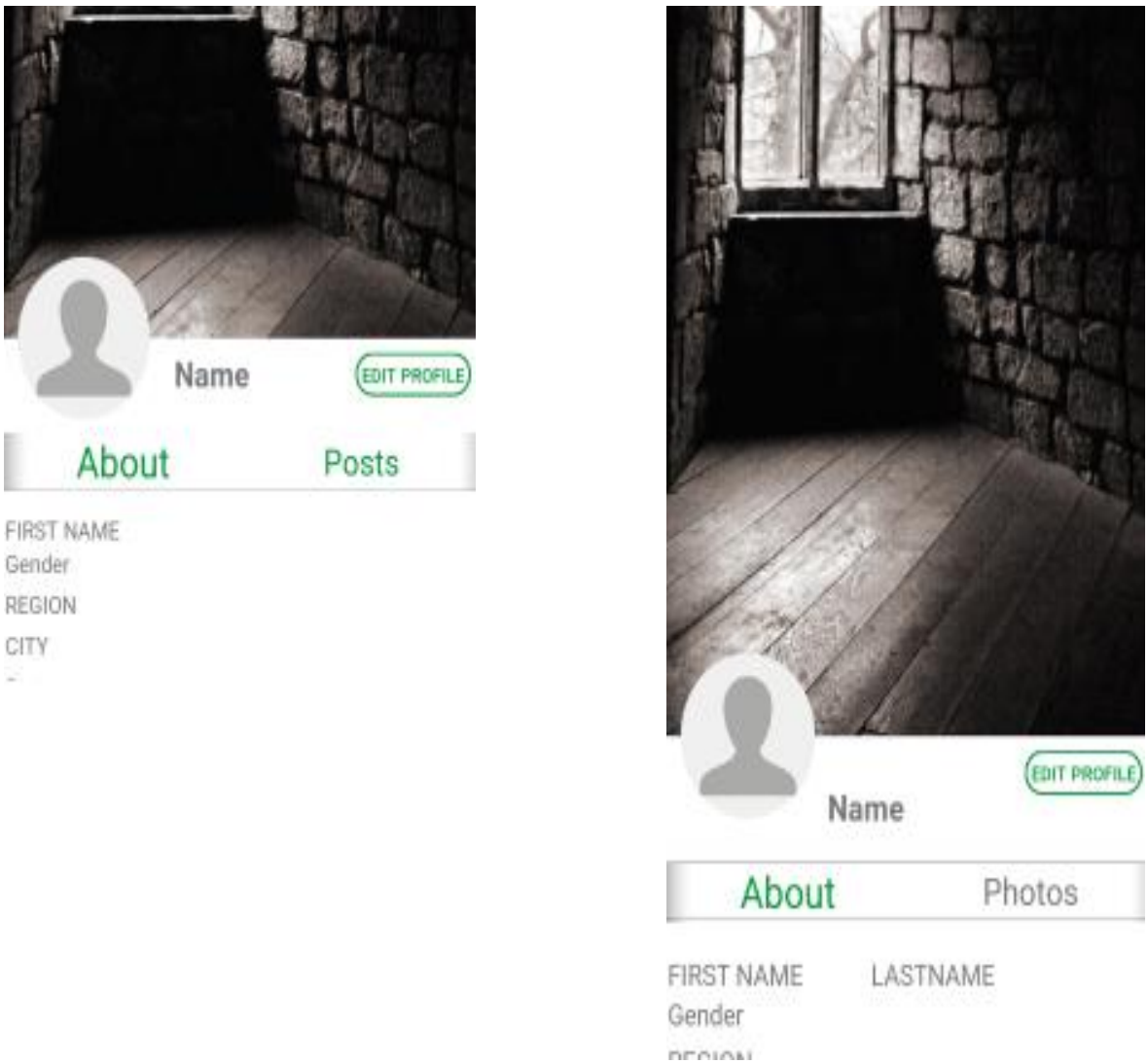

Figure 3. 13 Profile Screen Layout

3.14 Userprofile Activity

This activity is different from the profile fragment which gets the details of the current user but the “UserProfile” activity gets the details of other users or friends. The java codes which works behind this activity functions exactly like that of the profile fragments and more. This activity allows other users to view all the posts of a user, add the user as a friend and even unfriend the user all at the click of a button.

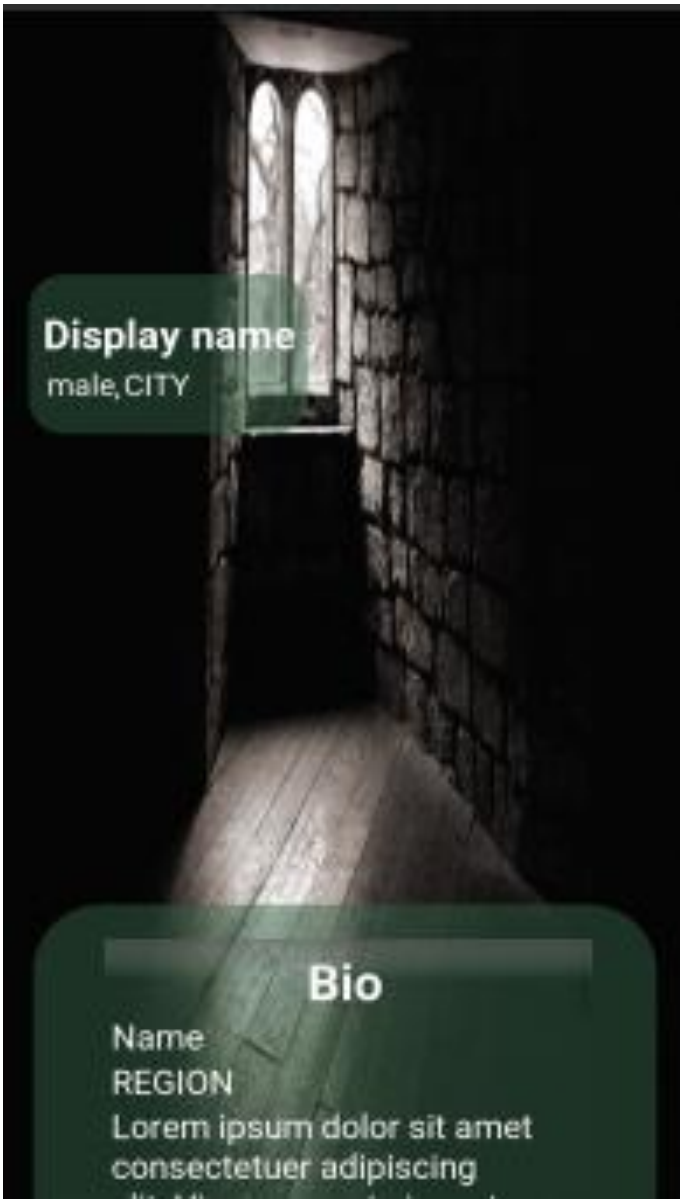

Figure 3. 14 UserProfile Screen Layout

3.15 Chats List Activity

The chats list activity, is an activity with its parent activity being the “MainActivityy” which shows all the users the current user has previously texted or has received a text from.

Figure 3. 15 Chats List Screen Layout

3.16 Chats Activity

This is the activity which enables users to actually send to text, voice notes and even make video and voice calls to other users.

This activity allows the user to send text messages to other users by clicking the “send” button which only shows after a character has been typed. Also, it allows users for make calls to the users by clicking the voice call or video call buttons. It allows users to send voice notes, live videos and picture as well as pictures saved in the phone’s gallery to other users. All these happen at the touch of a button.

Figure 3. 16 Chats Activity Screen Layout

3.17 Special Classes and Methods

The following are model classes and methods which perform specific task vital to the smooth running of the application.

3.17.1 Messages.class

This is a model class of the details of the messages stored in the database. This has been specially created to make it easy to get messages, time, recipients, sender etc. of the message with the help of the methods in the class.

3.17.2 Posts.class

This is a model class of the details of the posts stored in the database. This has been specially created to make it easy to get posts, time, owner, number of likes and comments etc. of the post with the help of the methods in the class.

3.17.3 PagerViewAdapter1.class

This class is a special adapter which makes the “viewpager” function without any complications.

3.17.4 MessageAdapter.class

This adapter class makes it possible for the heterogeneous layout of the chats activity to function very well. This adapter class could be termed as the “engine of the chats activity”.

3.17.5 Users.class

This is a model class of the details of the users stored in the database. This has been specially created to make it easy to get first names, last names, gender, time, recipients, sender etc. of the users with the help of the methods in the class.

CHAPTER 4

RESULTS AND DISCUSSIONS

This chapter talks about the outcomes of our findings or research. During our research we made some head ways as well as some shortcomings. The aim of our research was ahead up which a social networking application which would provide a lot more flexibility when compared with other social networking applications.

Flexibility in the sense that the application form allows user to:

- Speak with other users as all the social media marketing applications do.
- Enable easy monetary transactions by embedding APIs like the "Transfer Wise" API which would allow users to send and receive money across the planet, the "Venmo" API which would allow users of Yɛnkɔnecti in the US to make same transactions all the Venmo users are allowed to. That is, they would manage to split bills amongst friends and pay them.
- Provide platform for quick use of current news around the planet by linking the application form to the internationally recognized news agency called "ALJAZEERA".
- Give a platform for buying and selling by linking users to a favorite electronic market places in Ghana called "TONATON".
- Able to provide users with directions to local restaurants, fuel stations and hospitals inside their vicinity with the touch of a button.

We were able to accomplish about 90 percent of our goals and they are as follows:

1. We were able to come up with a social media application which would allow users to keep in touch with other users on the platform through texting, sending and receiving images and video etc. Also, users were to

meet new people, creating and fostering relationships with them on the platform.

2. The application allowed the transfer of money via the “Transfer Wise” platform embedded in the application.
3. Users were able to the newest news circulating around the globe from the “ALJAZEERA” platform embedded in the application.
4. Users were able to conduct e-commerce transactions like buying and selling as a result of “TONATON” platform embedded in the application.
5. Users were able to locate nearby fuel stations, hospitals and local restaurants as a result of google maps API embedded in the application.

Unfortunately, we were not able to implement the “VENMO” API like we wished to because, to use the API our location should be in the “USA” which wasn't and we did not need implement an API inside our application if we'd not manage to use and test for errors. This was the only objective we said we'd implement but we couldn't do.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This study centered on the development of an android social media application, an application that supports networking and is of importance. The development of the Yɛnkɔnecti was created solely on the Android environment with which the front end of was developed with XML and Java whilst itd back end was developed with Firebase and the Google Maps API framework.

The application was not only developed for socialization but can also serve as a platform that facilitates monetary transactions and other electronic commerce activities, a platform that provides users with current news from across the globe and ability to provide users with locations of nearby hospitals, restaurants and fuel stations.

5.2 Recommendations

The following was recommended for the study.

1. Since we were not able to implement the “VENMO” API like we wished to because to use the API our location should be in the “USA”, our future work would include a work around which would perform similar functions as Venmo does.
2. Our future work should also enable users to locate friends and families on the map.
3. Work should be done on the video and voice call aspects when it comes to communication.
4. Work should be done on the monetary transaction part of the application to reduce the number of third-party APIs embedded in the app just to increase security.

REFERENCES

1. Adam Sinicki (December 24, 2017). “Android Development” https://www.androidauthority.com/develop-android-apps-languages-learn-391008/ . Retrieved January 17, 2019.
2. Aggarwal Y.R., (1998), “Statistical Methods”, Sterling Publications Pvt. ltd., New Delhi.
3. "Android – Google Mobile Services". Android. Retrieved January 17, 2019.
4. "Android Language Breakdown". Open Hub. October 25, 2017. Archived from the original on December 14, 2017. Retrieved January 17, 2019.
5. "Android 7.0 Nougat". Archived from the original on August 22, 2016.Retrieved January 17, 2019.
6. Best J.W., (2003), “Research in Education”, Ninth Edition, Prentice Hall of India, New Delhi.
7. Drew Hendricks (May 8, 2013). “Complete History Of Social Media: Then and Now”. https://smallbiztrends.com/2013/05/the-complete-history-of-social-mediainfographic.html. Retrieved November 18, 2018.
8. Christian De Looper (October 23, 2018). “Android 1.0 to Android 9.0, here’s how Google’s OS evolved over a decade” https://www.digitaltrends.com/mobile/android-version-history/. Retrieved January 17, 2019.
9. Christian De Looper (November 23, 2018). “Enlist in the Android Army: A beginner’s guide to Android” https://www.digitaltrends.com/mobile/guide-to-android/ . Retrieved January 17, 2019.
10. "History and Different Types of Social Media". http://scalar.usc.edu/works/everything-you-always-wanted-to-know-about-social-media- but-were-too-afraid-to-ask/history-and-different-types-of-social-media

University of Southern California. Retrieved November 18, 2018

11. Jr Raphael (August 7, 2018). “Android versions: A living history from 1.0 to Pie” https://www.computerworld.com/article/3235946/android/android-versions-a-living-history-from-1-0-to-today. html. Retrieved January 17, 2019.
12. Mangal S.K., (2005), “Statistics in Psychology and Education”, Second Edition, Prentice Hall of India, New Delhi.
13. Mahmud Af Nahin (January 9, 2019). “Android 2.0 Eclair — Google Maps Navigation, Html5 Browser Support, And New Lock Screen (2009)” https://c.mi.com/thread-1789474-1-0.html. Retrieved January 17, 2019.
14. Morrill, Dan (September 23, 2008). "Announcing the Android 1.0 SDK, release 1". Android Developers Blog. Google. Archived from the original on March 5, 2017. Retrieved January 17, 2019.
15. Nicolas Vega (JULY 23, 2018). “SnapChat is discontinuing SnapCash function” google.com/amp/s/nypost.com/2018/07/23/snapchat-is-discontinuing-its-snapcashfunction/amp/. Retrieved November 18, 2018
16. Nathaniel Smithson (September 14, 2018). “Facebook Inc.’S Mission Statement & Vision Statement (An Analysis)” panmore.com/Facebook-inc-vision-statement-mission-statement.Retrieved November 18, 2018.
17. Percival Fred (1994), “A handbook of Educational Technology”, Cogan pages Publishing House, London.
18. Supriya (April 3, 2017). “Evolution of Android” https://www.techjini.com/blog/evolution-of-android/. Retrieved January 17, 2019.
19. Virginia Strvea (July 23, 2018). “SnapChat is saying goodbye to SnapCash” phillyvoice.com/snapchat-ending-snapcash-venmo-zelle-payment-service-/amp. Retrieved November 18, 2018

20. “What is Venmo” help.venmo.com/hc/en-us/articles/221011388-What-is-Venmo-Venmo. November 16, 2018. Retrieved November 17, 2018.
21. Whatsapp Mobile Application https://www.whatsapp.com/about. Retrieved November 18, 2018.
22. Wikipedia https://en.wikipedia.org/wiki/Social_media. Retrieved November 18, 2018.

APPENDIX A

The codes below are the xml codes for the splash screen activity.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SplashScreenActivity"
android:orientation="vertical"
android:background="@drawable/splash1">
</RelativeLayout>
```

Below are the java codes for the welcome page activity:

```
public class WelcomePageActivity extends AppCompatActivity implements
View.OnClickListener{
private Button siginb, signupb;
static WelcomePageActivity activityA;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.welcome_page);
siginb = findViewById(R.id.login);
signupb = findViewById(R.id.Sign_up);
siginb.setOnClickListener(this);
```

APPENDIX A

```
signupb.setOnClickListener(this);
activityA = this;
}
public static WelcomePageActivity getInstance(){
return activityA;
}
@Override
public void onClick(View v) {
switch (v.getId()) {
case R.id.login:
Intent start1 = new Intent(WelcomePageActivity.this, LoginPageActivity.class);
startActivity(start1);
break;
case R.id.Sign_up:
Intent start2 = new
Intent(WelcomePageActivity.this,RegistrationPageActivity.class);
startActivity(start2);
break;
default:
break;
}}}
```

APPENDIX A

Below are the java codes for the login page activity:

```
public class LoginPageActivity extends AppCompatActivity implements
View.OnClickListener {
private TextInputLayout em, pass;
private TextView reset;
private Button log;private FirebaseAuth mAuth;
private FirebaseUser mCurrentUser;
private FirebaseFirestore mFirestore;
private CollectionReference mUserinfo;
private String Email, Password;
private ProgressDialog note;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.login_page);
em = findViewById(R.id.mail_login_in);
pass = findViewById(R.id.password_login_in);
reset = findViewById(R.id.nb);
reset.setOnClickListener(this);
log = findViewById(R.id.login_btn);
log.setOnClickListener(this);
mFirestore = FirebaseFirestore.getInstance();
mAuth = FirebaseAuth.getInstance();
mUserinfo = mFirestore.collection("Users");
```

APPENDIX A

```
note = new ProgressDialog(this);
}
public void Sigin() {
Email = em.getEditText().getText().toString().trim();
Password = pass.getEditText().getText().toString().trim();
note.setMessage("Please wait....signing in");
note.show();
String Token = FirebaseInstanceId.getInstance().getToken();
final Map<String,Object> tokken = new HashMap<>();
tokken.put("device_token",Token);
mAuth.signInWithEmailAndPassword(Email, Password).
addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
@Override
public void onComplete(@NonNull Task<AuthResult> task) {
if (task.isSuccessful()) {
mCurrentUser = mAuth.getCurrentUser();
mUserinfo.document(mCurrentUser.getUid()).update(tokken).addOnSuccessListener
(new OnSuccessListener<Void>() {
@Override
public void onSuccess(Void aVoid) {
Intent start1 = new
Intent(LoginPageActivity.this,MainActivityy.class);startActivity(start1);
finish();
}});} else {
```

APPENDIX A

```
Toast.makeText(LoginPageActivity.this, "So error signing in!!"
, Toast.LENGTH_SHORT).show();
} note.dismiss();}});}
@Override
public void onClick(View v) {
switch (v.getId()) {
case R.id.login_btn:
Sigin();
WelcomePageActivity.getInstance().finish();
break;
case R.id.nb:
Intent star
Intent(LoginPageActivity.this,resetting.class);
startActivity(star);
break;
default:
break;
}}}
```

Below are the XML codes for the registration page activity describing how the registration activity should look ;

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools=http://schemas.android.com/tools
```

APPENDIX A

```
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".RegistrationPageActivity"
android:background="@color/colorPrimary1"><include
android:id="@+id/RegisTBar"
layout="@layout/customizedtoolbar" />
<ScrollView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_below="@id/RegisTBar">
<LinearLayout
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
<de.hdodenhof.circleimageview.CircleImageView
android:layout_width="150dp"
android:layout_height="150dp"
android:id="@+id/dpooo"
android:layout_marginStart="120dp"
android:src="@drawable/default_avatar"
android:layout_marginTop="10dp"/>
<android.support.design.widget.TextInputLayout
android:id="@+id/User"
android:layout_width="match_parent"
```

APPENDIX A

```
android:layout_height="wrap_content"
android:layout_below="@id/dpooo"
android:layout_marginLeft="10dp"
android:layout_marginRight="10dp"
android:layout_marginTop="10dp">
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Username"
android:singleLine="true"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/mailer"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/dpooo"
android:layout_marginLeft="10dp"
android:layout_marginRight="10dp"
android:layout_marginTop="10dp">
<EditText
android:singleLine="true"
android:layout_width="match_parent"
android:layout_height="wrap_content"
```

APPENDIX A

```
android:hint="@string/First"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/surname"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/mailer"android:layout_margin="10dp">
<EditText
android:singleLine="true"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/Sur"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/gender"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/mailer"
android:layout_margin="10dp">
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
```

APPENDIX A

```
android:hint="Gender"
android:singleLine="true"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/DOB_main_registration_in"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/surname"
android:layout_margin="10dp">
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:singleLine="true"
android:hint="@string/DOB"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/email_registration_main_in"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/DOB_main_registration_in"
android:layout_margin="10dp">
<EditText
```

APPENDIX A

```
android:singleLine="true"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/email"
android:inputType="textEmailAddress"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/password_registration_in"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/email_registration_main_in"android:layout_margin="1
0dp">
<EditText
android:singleLine="true"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/password"
android:inputType="textPassword"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/confirmation_in"
android:layout_width="match_parent"
```

APPENDIX A

```
android:layout_height="wrap_content"
android:layout_below="@id/password_registration_in"
android:layout_margin="10dp">
<EditText
android:singleLine="true"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/confirm"
android:inputType="textPassword"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
android:id="@+id/region_in"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/confirmation_in"
android:layout_margin="10dp">
<EditText
android:singleLine="true"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="@string/region"
android:textColor="@color/black"/>
</android.support.design.widget.TextInputLayout>
```

APPENDIX A

```
<android.support.design.widget.TextInputLayout
android:id="@+id/city_in"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/region_in"
android:layout_margin="10dp">
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:singleLine="true"
android:hint="@string/city" />
</android.support.design.widget.TextInputLayout>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal">
<Buttonandroid:id="@+id/registrationdone"
android:layout_width="157dp"
android:layout_height="wrap_content"
android:layout_below="@id/city_in"
android:layout_marginLeft="30dp"
android:background="@drawable/button_border5"
android:text="@string/save"
android:textColor="@color/tol"
```

APPENDIX A

```
android:textSize="10pt" />
<Button
android:id="@+id/cancellation"
android:layout_width="139dp"
android:layout_height="wrap_content"
android:layout_below="@id/city_in"
android:layout_marginLeft="8dp"
android:layout_toRightOf="@id/registrationdone"
android:background="@drawable/button_border5"
android:text="@string/cancel"
android:textColor="@color/tol"
android:textSize="10pt" />
</LinearLayout>
</LinearLayout>
</ScrollView>
</RelativeLayout>
```

Below are the java codes which determines the operation of the views in the reset password activity:

```
public class resetting extends AppCompatActivity {
private EditText email;
private Button reset;
private String mail;
private FirebaseAuth resetter;
@Override
```

APPENDIX A

```
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_resetting);
email = (EditText) findViewById(R.id.yo);
reset = (Button) findViewById(R.id.resett);
reset.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Reset();}});
resetter = FirebaseAuth.getInstance(); }
public void Reset() {
mail = email.getText().toString().trim();
resetter.sendPasswordResetEmail(mail)
.addOnCompleteListener(this, new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if (task.isSuccessful()) {
Toast.makeText(resetting.this, "A reset link has been sent to your" +
"mail",
Toast.LENGTH_SHORT).show();}}});}}
```

Below are the xml codes which describes the layout of this activity :

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app=http://schemas.android.com/apk/res-auto
```

APPENDIX A

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="fill_parent"
android:background="@color/colorPrimaryDark1"
android:orientation="vertical"
tools:context=".MainActivityy">
<RelativeLayout
android:id="@+id/topPanel"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:background="@drawable/border"
android:padding="5dp">
<ImageButton
android:id="@+id/chats"
android:layout_width="52dp"
android:layout_height="wrap_content"
android:layout_marginStart="5dp"
android:layout_weight="1"
android:background="@null"
android:scaleType="centerInside"
android:src="@drawable/chat" />
<TextView
android:id="@+id/pageName"
android:layout_width="wrap_content"
```

APPENDIX A

```
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:text="Page Name"
android:textSize="21dp"
android:textStyle="bold" />
<ImageButton
android:id="@+id/sign_out1"
android:layout_width="50dp"
android:layout_height="35dp"
android:layout_alignParentEnd="true"
android:layout_marginEnd="9dp"
android:background="@null"
android:scaleType="fitCenter"
android:src="@drawable/sign_out" />
</RelativeLayout>
<android.support.v4.view.ViewPager
android:id="@+id/mainPager"
android:layout_width="match_parent"
android:layout_height="525dp"
android:layout_above="@id/tabs_layout1"
android:layout_below="@id/topPanel" /><LinearLayout
android:id="@+id/tabs_layout1"
android:layout_width="match_parent"
```

APPENDIX A

```
android:layout_height="45dp"
android:layout_alignParentBottom="true"
android:background="@color/white3"
android:orientation="horizontal"
android:visibility="visible"
android:padding="5dp">
<de.hdodenhof.circleimageview.CircleImageView
android:id="@+id/Home"
android:layout_width="40dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:gravity="center"
android:src="@drawable/ic_home"
app:civ_border_color="@color/white"
app:civ_border_width="3dp"/>
<de.hdodenhof.circleimageview.CircleImageView
android:id="@+id/community"
android:layout_width="40dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:gravity="center"
android:src="@drawable/search2"
app:civ_border_color="@color/white"
app:civ_border_width="3dp"/>
```

APPENDIX A

```
<de.hdodenhof.circleimageview.CircleImageView
android:id="@+id/connectiWorld"
android:layout_width="40dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:gravity="center"
android:src="@drawable/yenkworld"
app:civ_border_color="@color/white"
app:civ_border_width="3dp"/>
<de.hdodenhof.circleimageview.CircleImageView
android:id="@+id/friends"
android:layout_width="40dp"
android:layout_height="match_parent"
android:layout_weight="1"
app:civ_border_color="@color/white"
app:civ_border_width="3dp"
android:src="@drawable/friend"
/>
<de.hdodenhof.circleimageview.CircleImageView
android:id="@+id/Profileg"android:layout_width="40dp"
android:layout_height="match_parent"
android:layout_weight="1"
android:gravity="center"
app:civ_border_color="@color/white"
```

APPENDIX A

```
app:civ_border_width="3dp"
android:src="@drawable/default_avatar" />
</LinearLayout>
</RelativeLayout>
```

Below are the java codes for the YenkonnetiWorld activity:

```
public class YenkonetiWorld extends Fragment implements View.OnClickListener {
private CardView Money,Trade,News,Nearby;
public YenkonetiWorld() {}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle savedInstanceState) {
View v = inflater.inflate(R.layout.fragment_yenkoneti_world, container, false);
Money = v.findViewById(R.id.Monetory);
Trade = v.findViewById(R.id.Trade);News = v.findViewById(R.id.news);
Nearby = v.findViewById(R.id.Locator);
Money.setOnClickListener(this);
Trade.setOnClickListener(this);
News.setOnClickListener(this);
Nearby.setOnClickListener(this);
return v;}
@Override
public void onClick(View view) {
switch (view.getId()){
case R.id.Monetory:
```

APPENDIX A

```
Intent Financial = new Intent(getContext(), MoneyTransaction.class);
startActivity(Financial);
break;
case R.id.Trade:
Intent commerce = new Intent(getContext(),Trade.class);
startActivity(commerce);
break;
case R.id.news:
Intent newsFeed = new Intent(getContext(),Feed.class);
startActivity(newsFeed);
break;
case R.id.Locator:
Intent locale = new Intent(getContext(),NearbyLoc.class);
startActivity(locale);
break; }}}
```