Apprize: A Social Platform for Self-Wellness Using Decision Tree Algorithm

A Thesis
Presented to
The Faculty of the College of Computer Studies
Lyceum of the Philippines University – Batangas

In Partial Fulfillment
Of the Requirements for the Degree
Bachelor of Science in Computer Science
Specialized in Mobile Application Development

By:
John Christopher L. Austria
Jazril Karlo P. Bagui
Mark Kenneth B. Dalisay
Nick Paolo Lodana

June 2020

APPROVAL SHEET

In partial fulfillment of the requirements for the degree Bachelor of Science in Computer Science (Specialized in Mobile Application Development), this thesis entitled **"Apprize: A Social Platform for Self-Wellness Using Decision Tree Algorithm"** has been prepared and submitted by **JOHN CHRISTOPHER AUSTRIA, JAZRIL KARLO BAGUI, MARK KENNETH DALISAY**, and **NICK PAOLO LODANA.**

Roselie B. Alday, MCS, Ph.D. Candidate
Adviser

Defended in an oral examination before a duty constituted panel with a grade of _____.

Maria Cristina M. Ramos, MSCS
Chairman

Elaine Joy J. Ilao, MAITE
Member

Von Derick G. Ebreo
Member

Dr. Arnie Christian D. Villena
Member

Accepted in partial fulfillment of the requirements for the degree Bachelor of Science in Computer Science (Specialized in Mobile Application Development).

Roselie B. Alday, MCS, Ph.D. Candidate
Dean, College of Computer Studies

TABLE OF CONTENTS

List of Figures

ACKNOWLEDGEMENT

The authors would sincerely like to extend their most profound gratefulness to the individuals who enormously contributed to the accomplishment of this research project;

First of all, to our ever-cherished family, for their incredible help monetarily and ethically, and for their unconditional love and boundless motivation that inspired the researchers all throughout the process;

To our dearest thesis adviser and college dean, Mrs. Roselie Alday, for uplifting our spirits with her guidance, words of encouragement, heartening support, and patience in reviewing the researcher's work;

To our enthusiastic research panelists, Mrs. Maria Cristina Ramos, Ms. Elaine Ilan, and Sir Von Derick Ebreo, for being whole-heartedly open, understanding, and approachable in proposing improvements to the project's progress;

To Mrs. Merlinda Pesigan and Sir Aries Mendoza, for greatly helping the researchers especially in the mobile application development by sharing their insights and expertise in their respective fields;

To our inspirational colleagues and friends, for their morale boosters and unwavering support while completing the research;

Above all, to our Almighty Father, for giving vast blessings, great well-being, perseverance and wisdom to develop this thesis project.

John Christopher

Jazril Karlo

Mark Kenneth

Nick Paolo

DEDICATION

APPRIZE is dedicated to all our love ones who were always there believing in us from the very first day, nurturing us in the process of character growth, supporting us in reaching our dreams, guiding us in our life journey, and being our backbones while we constantly pursue the best versions of ourselves;

To all our noble college professors and mentors who shaped us to be life-ready mentally and spiritually by sharing their knowledge, philosophies, and abilities that greatly touched our lives;

To the individuals who are constantly facing their own battles, may they keep on enduring, discover the magnificence of life, and hope for more brilliant days to come, as they seek purpose and inspire other people;

To every young technology enthusiast, who are continuously learning and grinding to make the world a better place through valuable innovations and creative outputs;

And finally, to our dear selves, as we continue to take the lead and pursue excellence, and trailblaze the paths we want to conquer.

To God be all the glory!

Austria

Bagui

Dalisay

Lodana

Apprize: A Social Platform for Self-Wellness Using Decision Tree Algorithm

John Christopher L. Austria
Lyceum of the Philippines University
Capitol Site, Batangas City
09266252662
jayceaustria@lpubatangas.edu.ph

Mark Kenneth B. Dalisay
Lyceum of the Philippines University
Capitol Site, Batangas City
09164046730
kennethdalisay@lpubatangas.edu.ph

Jazril Karlo P. Bagui
Lyceum of the Philippines University
Capitol Site, Batangas City
09473979601
jazrilbagui@lpubatangas.edu.ph

Nick Paolo Lodana
Lyceum of the Philippines University
Capitol Site, Batangas City
09206917716
nickpaololodana@lpubatangas.edu.ph

ABSTRACT

Self-wellness can be easily neglected these days especially with the high usage of technology and diverse devices every day, either for work or for leisure. These can sometimes lead to some mental and physical health issues that may be severe if not taken care of. This research is a mobile application that can aid self-wellness. Social media applications can often be toxic to a person's mental health with people criticizing other people, but some applications can be relaxing and fun for other end-users. The effect of using these mobile applications can vary depending on the user. This thesis aids users by providing activities that can have positive effects on the user's mood or mental health. Through Apprize, users can see other users character façade by completing a brief personality test by tapping the Uplift feature where the Decision Tree Algorithm is utilized. The algorithm takes the user's set of answers as data and organize the result as their personality type. End-users can also engage in the activities in the Mood Lifter section of the mobile application. There are three main functions of the application such as Profile, Apprize and Uplift. In Profile, users can perceive their personal account information such as full name, birthdate and MBTI personality type. While Apprize provides three sections of features that can be explored – Mood Lifter, Personality Types in Birds personas, and Sentiments. The last function, Uplift, is the social platform section of the mobile application where end-users can view their friend matches and interactions.

Keywords*: Apprize, Decision Tree Algorithm, Mobile Application, Personality Types, Self-wellness, Social Platform*

1.0 INTRODUCTION

The complexity of this modern world is a difficult thing to understand, especially when it comes to human diversity, physical traits and abilities. Individual differences can be rooted from ancestral genetics, sexuality, ethnicity, personal upbringing, dissimilar cultures and environments, and life experiences, solidifying the fact that no person is exactly the same as the other. These unique features of every people make life more worthy of exploring and discovering but no one can ignore the fact that these variations also lead to misunderstandings and separations depending on situations and the ones involved. [1]

Isolation and loneliness are the results of these misapprehensions. People tend to disconnect themselves from situations they feel out of place and unwanted. The condition of being alone commonly define loneliness, which is actually a state of mind. It causes individuals to feel unwanted and empty. Individuals who feel lonely often crave physical human contact, but their state of mind hinders them to create genuine connections with others. Loneliness, as indicated by numerous specialists, isn't really about being distant from everyone else. Rather, in the event that you feel alone and disengaged, at that point that is the means by which loneliness plays into your perspective. [2] Different personality types also come into play when it comes to social isolation. Personality is explained as the totality of the physical, mental, emotional, and social characteristics of individuals. Every person's decisions and actions may or be based on their respective personalities and characters, basically, it makes people who they are. Personality is also a big factor in gaining self-confidence and improving one's self-esteem which are very significant to our day-to-day lives.[3] Poor character development and social exposure may cause a downside on one's personality that may result to solitude and isolation which may further aggregate loneliness or worse – may lead to self-harm and suicide.

The cure for loneliness or unwanted social disconnections are genuine personality understanding, social and physical contact, and intimacy. Some individuals don't have the courage to open up with their relatives and close friends, thus, others find empathy from strangers. Good thing, social networking sites and mobile applications were created.

People use these social networking sites for communication, information, and in building and maintaining relationships. Since one of the main purposes of social networking sites is to connect people from all over the world, learning to discover and find acquaintances is important to enjoy the benefits of a mobile application. Social media have made a great impact on the self-esteem of individuals, also, enhancing their network and quantity of friends.[4]

Self-esteem alludes to an individual's convictions about their own value and worth. It likewise has something to do with the emotions individuals experience that follow from their feeling of value or dishonor.

Confidence is significant in the light of the fact that it vigorously impacts individuals' decisions and choices. People with high self-esteem are additionally individuals who are roused to deal with themselves and to industriously endeavor towards the satisfaction of personal goals and aspirations. Individuals with lower confidence don't will in general see themselves as deserving of happy outcomes or capable of accomplishing them thus, tend to let significant things slide and to be less persevering and strong in terms of overcoming adversity. They may have the similar sorts of objectives as people with higher self-esteem, yet they are commonly less motivated to seek after them to their decisions. [5]

It is in this context that the researchers were determined to develop a social platform that will be of significant help to individuals who feel out-of-place and lonely to primarily boost their self-esteem.

"**Apprize"** is a mobile application and social platform that aids end-users to reconnect and positively socialize with new acquaintances. It is also an online mobile application that provides 30 simple self-wellness tasks and journal for personal sentiments, to uplift the end-user's mood positively. The mobile application will start with a Myers-Briggs personality examination that will determine the personality trait of the user. The respective personality test utilizes the Decision Tree Algorithm that helps the assessor give accurate results through a series of function branches the algorithm provides.

The 30 simple mood lifter tasks in the Apprize as a Mobile Application offers and may help an individual to bust bad vibes within 10 minutes or less with pleasurable yet beneficial activities such smiling, cuddling with someone, de-cluttering and many more. It also features a journal or note creator where the end-user may input personal thoughts left unsaid to at least lessen the mental burden of such. Apprize is also a social platform that provides a chat system with fellow end-users. With that being stated, Apprize doesn't only provide the end-user someone to talk to, but possibly someone who can also genuinely understand others based on the personality types. It is strictly for empowering individuals and spreading optimism. The mobile application will not provide clinical, professional and psychological results based on the end-user's task accomplishments and social experience with the application but rather, the mobile application will only be a leeway to initially understand and know oneself better through self-monitoring of thoughts, emotional and mental reactions from the activities, and connecting with fellow end-users. Apprize can't be utilized offline and was also developed for the Android Platform only with an OS version of Lollipop (5.0) up to the newest version. The RAM and internal or external memory must at least be 1GB. Meanwhile, the mobile device's screen orientation is set to portrait mode by default.

1.1 Objectives of the Study

The study entitled: "Apprize: A Social Platform Using Decision Tree Algorithm" aimed at attaining the following objectives:

1. To develop a mobile application that will aid the end-user to elevate oneself through personality test, performing basic mood lifting activities and socially connect to other people in a positive manner.

2. To utilize Decision Tree Algorithm in the development of the mobile application.

3. To use Android Studio in developing and designing the mobile application.

2.0 LITERATURE REVIEW

Nowadays, some teens and young adults often feel lonely and these states of mind are associated with different personalities of the people around them. Hence, understanding everyone's personality is a complex thing and it is uneasy finding a companion that doesn't suit their preferences when it comes to character. Social media helps to connect the world and various mobile applications can help solve these mental cases.

Promoting Public Health – Loneliness is Epidemic

According to a study from the British Red Cross, over nine million (9 million) adults in the U.K. feel they are left out or being isolated—that is about 1/5 of the country's population! Loneliness is seriously being considered a hazard to human health comparable to smoking and obesity. The mental state of loneliness or isolation affects everyone at some point, but persistent loneliness can become a severe dilemma that shortens lifespan and ruins physical health.

Dr. Vivek H. Murthy, former Surgeon General of the United States, wrote in an article in the Harvard Business Review, "Loneliness and weak social connections are associated with a reduction in lifespan similar to that, caused by smoking 15 cigarettes a day and even greater when associated by obesity." While the answer for the loneliness epidemic is multifaceted, urging individuals to assemble significant and commonly helpful associations is a positive development.

A research on social media interactions has indicated that dejection is more unavoidable in social orders and age bunches where web-based media usage is the most elevated. Various types of online media impact loneliness more than others. Yet, we can't simply accuse social media for damaging our relationships. [5]

The experience of loneliness is pervasive and detrimental. Character may impact individual perceptions of loneliness; however, relationships has not been satisfactorily inspected among minority maturing populaces. As an emotional inclination, dejection mirrors individuals' various examples in intuition, carrying on, and responding to situational factors. This, in turn, is affected by personality, depicted as one's dispositional characteristics and variations to the conditions. Henceforth, character may have an essential effect on the discernment and adapting of forlorn emotions.[6] Comprehending and empathizing each other's differences and personalities may help regain inclusion and boost one's self-esteem.

Understanding Personality Types

Personality typing is a system of categorizing individuals as per their inclinations to think and act in specific ways. Personality typing endeavors to locate the broadest, most significant manners by which individuals are extraordinary, and comprehend these distinctions by arranging individuals into meaningful groups.

The personality types described here were created by Isabel Briggs Myers and her mother, Katharine Briggs, in the 1960's. Myers and Briggs proposed that there were four key dimensions that could be used to categorize people: Introversion vs. Extraversion, Sensing vs. Intuition, Thinking vs. Feeling and Judging vs. Perceiving. Each of the four dimensions was described as a *dichotomy*, or an either/or choice between two styles of being.

Myers and Briggs depicted this as a "preference" and recommended that any individual ought to have the option to distinguish a favored style on every one of the four measurements. The whole of an individual's four preferred styles turns into their personality type. [7].

Alexandra Ehrenberg, a member of the Australian Psychology Society, and her team made a research about the role of personality and self-esteem in the usage of mobile applications. They found that introverts and people with low self-esteem are more eager to engage socially in mobile communications. Therefore, they find comfort in the infamous barriers created by mobile applications in opening up with other individuals.[8]

Mobile Application

Also called as mobile apps, it is a term used to portray Internet applications that executes on smartphones and other mobile devices. Mobile applications generally help end-users by interfacing them to Internet benefits more normally accessed on desktop or scratch pad PCs, or help them by making it simpler to utilize the Internet on their compact gadgets. A mobile application might be a portable Website bookmarking utility, a mobile-based instant texting customer, Gmail for portable gadgets, and numerous different applications.

It is devised with contemplations for the requests and requirements of gadgets and to exploit any specific capacities. A social application, for instance, may utilize the phone's local service provider.

Mobile applications are frequently designated according to whether they are web-based or local applications, which are made explicitly for a given platform. A third class, hybrid apps or crossover applications, joins components of both native and Web applications.[9]

Decision Tree Algorithm

Decision Tree Algorithms are considered to be one of the best and mostly used supervised methods. A decision can be compared with a flowchart that have a structure which each inside node represents a “test” on an attribute. It is also a support tool that uses a model of decisions and their possible consequences, or outcomes.

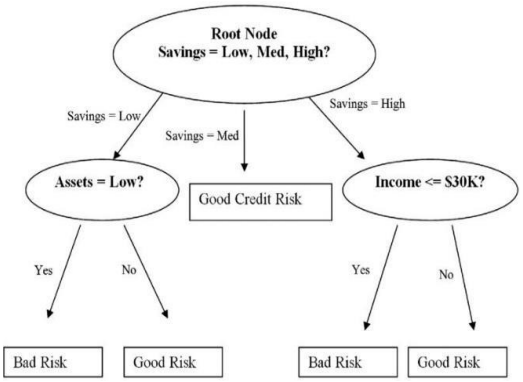

Figure 1. Algorithm Structure

Decision trees, random forest, gradient boosting are methods that are being used in all kinds of data manipulation and data science problems. [10] The researches utilized Decision Trees Algorithm since it gives a viable technique in dynamic cycles of making software decisions. It plainly spreads out the issue so the choices can be tested, permitting the researchers to analyze the potential results of a choice. It likewise gave a structure to evaluate and quantify the factors of the results and the feasibility of accomplishing them.

Firebase Database

When incorporated with Firebase Authentication, developers can characterize who access what information, and how they can get to its respective data. The Realtime Database is a NoSQL database and as such has various advancements and functionality contrasted with a social and relational data set. Firebase gives end-users functionality like analytics, databases, messaging and crash reporting so you can move rapidly and center around your clients. Firebase is based on Google framework and scales automatically, for even the most advanced applications. [11] Since Apprize will be developed as a social platform that will be utilized with various accounts, the mobile application is in need of a database that will contain the all the users' data. The mobile application will exploit Firebase database for it is well-suited in an Android platform and features cloud functions and storage.

Mobile Applications that Promote Positivity

Happify, Gratitude Journal – The Life Changing Application, Calm, Mindful Moon App, Pozify, Positive Thinking, YouTube, Realifex (Real Life Change), Smiling Mind some of these permits you to follow the optimistic energy in your life by following certain goals and focusing on the cheerful and positive parts of life, a few offers positive and enabling every day messages, while some presents positive news. Innovation is an advantageous method to get to programs that promote inspiration. Focusing on the great isn't just beneficial for your well-being and prosperity but at the same time it's useful for the people around you. [12] The promotion of optimism is also one of the purposes of the researchers' project. Apprize comprises self-therapeutic activities that aims to brighten an individual's mood and feelings.

Applications that Promote Mental Health Awareness

What's Up

What's Up is an application developed to give supportive devices for overseeing discouragement and depression. The application depends on standards of Cognitive Behavior Therapy (CBT) and incorporates numerous highlights. What's Up doesn't have

alternatives for sharing information or in any case remembering emotional well-being suppliers for the application, yet it might be a helpful guide for people with mellow-moderate depression who do not have access to conventional treatments [13]. In comparison with the researcher's project, What's Up and Apprize mutually aims to lessen mental negativity. Apprize just edges What's Up in a context that the mobile application is also a social platform.

Insight Timer

Insight Timer is a mobile application and online network for intervention. While its focal capacity is a straightforward clock, the application incorporates social features, and educators distribute guided meditations, music and talks that are gotten to by means of the application. It is the most well-known free contemplation application on significant platforms, with more than four million end-users. It was initially designed by Brad Fullmer starting in 2008 and bought by Australian siblings Christopher and Nicholas Plowman in April 2015. The mobile application made Time magazine's list of 50 best applications for 2016 [14]. Apprize is similar to Insight Timer in a way both mobile applications provide therapeutical activities for simple meditation. Though Apprize is also composed of activities that promote positivity in an individual, it features endeavors that can be done socially with a fellow user.

Talkspace

Talkspace is an online therapy startup that offers reasonable and classified treatment with organization of expert and authorized therapists wherever customers need assistance. Talkspace's main goal is to make a billion people blissful. With Unlimited Messaging Therapy, clients have access to an authorized, licensed and proficient advisor - no arrangement. [15] While Apprize doesn't provide authentic professional care, the mobile application brings socialization and support in a different variety. Since Apprize will display an individual's personality type in its profile, it aims to merge people with common personalities for better mingling and understanding.

Code blue

Code Blue is a simple smartphone application that goes about as a versatile help group. Developed to assist the youth get help when they most need it, it lets you select an aid team who, on accepting a 'Code Blue', will offer prompt efficient help by means of text, phone activity, or by appearing. [16] Help is a key term that both Code Blue and Apprize campaigns. Apprize is a mobile application that will provide self-help deeds and assistance from friends that was designed to aid and support the respective individual.

3.0 METHOD

Mobile Application Development Life Cycle

Java is the programming language used by the programmers to develop the application, which is object-oriented, developed by James Goslin at Sun Microsystem. It is machine independent because instead of generating native machine code, byte-codes are generated. The compile byte-codes are generated through the Java interpreter. (Sartipi, 1996)

The researchers are instructed to emphasize the goal and purpose of the thesis venture. Hence, the researchers rationalize the project's feasibility by formulating a systematic process and procedures that will be utilized in the development of the mobile application. This also includes determining the objectives and requirements to guarantee that bases are comprised. The aim and objectives of the project was attained with the procedural directions given by the used model – the Mobile Application Development Life Cycle (MADLC).

The purpose of this model is to help the researchers in developing the mobile application and to track its development phases. It guided the researchers to determine whether the mobile application can be successfully created or not. The researchers were able to critically analyze the issues and problems that appeared during the mobile application's development phase through the model.

5 Phases of Mobile Application Development Life Cycle

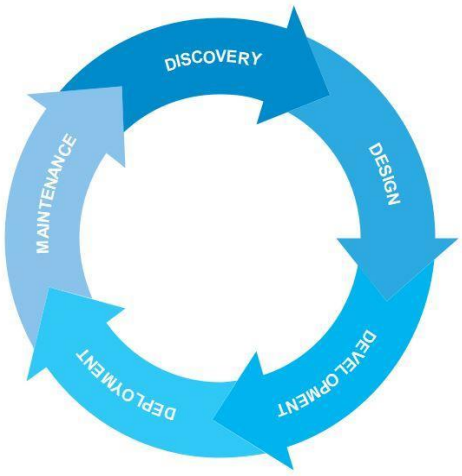

Figure 2. MADLC

Discovery Phase

The researchers conducted an open conversation and searched for current problems of people specifically on mental health problems. After brainstorming, they conceptualized a mobile application that aims to aid with the growing mental health issues, specifically with loneliness, and they planned a strategic way to develop the prototype. They gathered information by analyzing and understanding articles, reviews and statistics about the leading causes of mental health problems and its worldwide solutions. The researchers also referred their intended study to a professional Psychology practitioner to further understand their topic and develop it in an acceptable way. After that procedure, the proponents presented a proposal document consulted initially by their previous research adviser, Mrs. Melody Dimaano, who was later substituted by Dean Roselie Alday, who approved the proposed mobile application as well.

Design Phase

After the proposed document was approved, the researchers articulated their schedules in order to organize the time frame of the mobile application development process. They also identified all the possible risks and problems of the project in order to manage it effectively. The objectives, features, functionalities, target users and limitations of the study were also initiated during this phase.

The researchers formulated a design that suites the ideal visuals of the mobile application. They utilized Adobe Illustrator in designing the logo while Adobe XD was used in constructing the

initial user interface designs of the mobile application. The researchers also studied and chose Android Studio as the development tool and Google Firebase for cloud database handling. After determining and applying the softwares appropriate for the development of the project, the researchers visualize the functionalities, flowcharts and systematic plans of the venture. They ensured that the layouts were engaging and easy to use.

Development and Testing Phase

The researchers initiated the development of the application after successfully dispensing with the design stage. The usage of the Decision Tree Algorithm in handling and manipulating the data included inside the mobile application. The questions and the choices are stored in an array that can be called once the user selects an answer. The questions are classified as the root nodes and the succeeding questions would be the internal node. Every choice that the user picks, the node would split into nodes.

By running series of tests, the researchers are now ready to move into the programming and coding proper. The development cycle is cyclical and iterative. The process undergoes series of testing so that errors and bugs in the program may be detected to build up a corresponding solution. The phase helped the researchers to determine the things that are essential in order to avoid difficulties and casualties. The mobile application should be tested for usability, performance, compatibility and security.

The researchers used Android Studio in developing the mobile application and utilized Google Firebase for its database. The researchers studied different application tutorials for the development of the mobile application and applied the acquired knowledge in coding the mobile application's features and functions. The researchers used Google Firebase for handling the user accounts, pictures and text for the profile, notes created in the sentiments activity of the mobile application and the messages in the chat activity for the display of the conversation.

The researchers requested some individuals to test the mobile application for feedbacks and suggestions that is used in the maintenance and updates.

Deployment Phase

At this stage, the mobile application is ready to be released in the digital world after undergoing rigorous development and testing. The target of this phase is to generate end-users of the project as many as possible to gather suggestions and feedbacks that are essential for the improvement of the mobile application. The researchers attempted to upload the finished mobile application on Google Play Store for deployment in order for it to be accessed by the public. The project should conform with ISO 9126 standard, a criterion for the evaluation of software that is known internationally. The researches should primarily establish the initial portion of the standard which is ISO 9126-1 that emphasizes the quality of software in a structured set of factors: Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability.[17] By complying with the standards, the researchers were able to: recognize if the mobile application is effective and helpful to the targeted end-users and

identify additional recommendations for the project.

Maintenance and Update Phase

After the deployment phase, the researchers analyzed the performance of their application for maintenance and gather the feedbacks, suggestions, and bugs for updates in this phase. [18] The phase allowed the researchers to discover and plan new ways of improving and polishing the mobile application.

Flowcharts

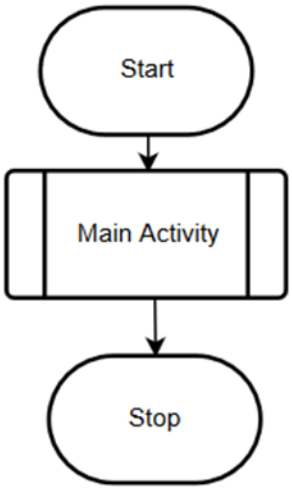

Figure 3. Apprize Flowchart

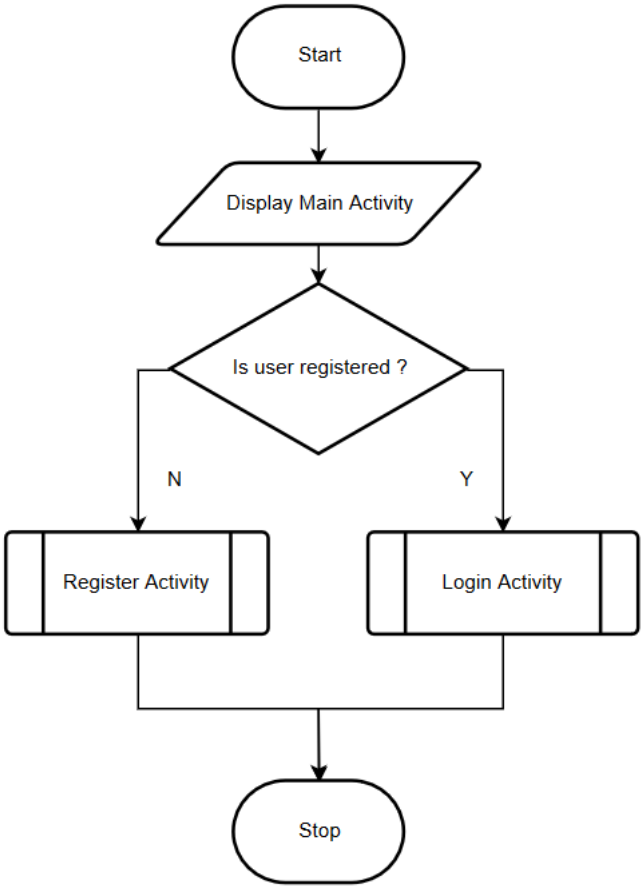

Figure 4. Main Activity

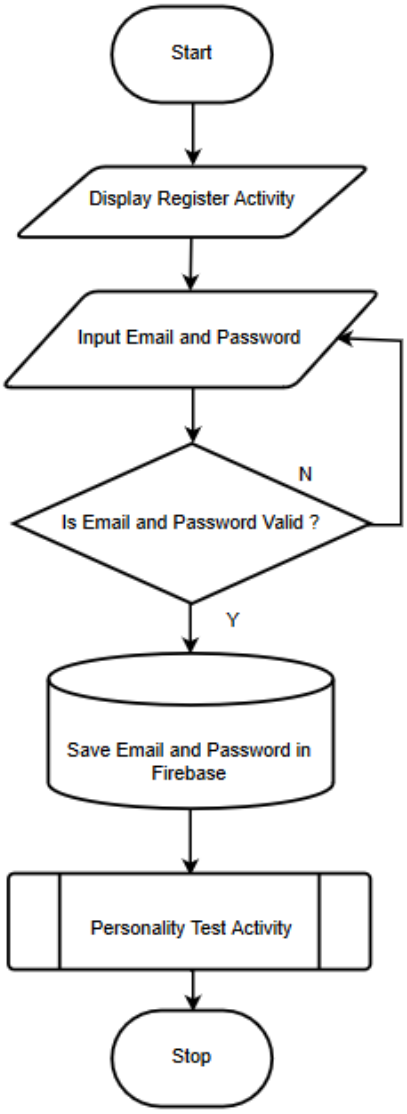

Figure 5. Register Activity

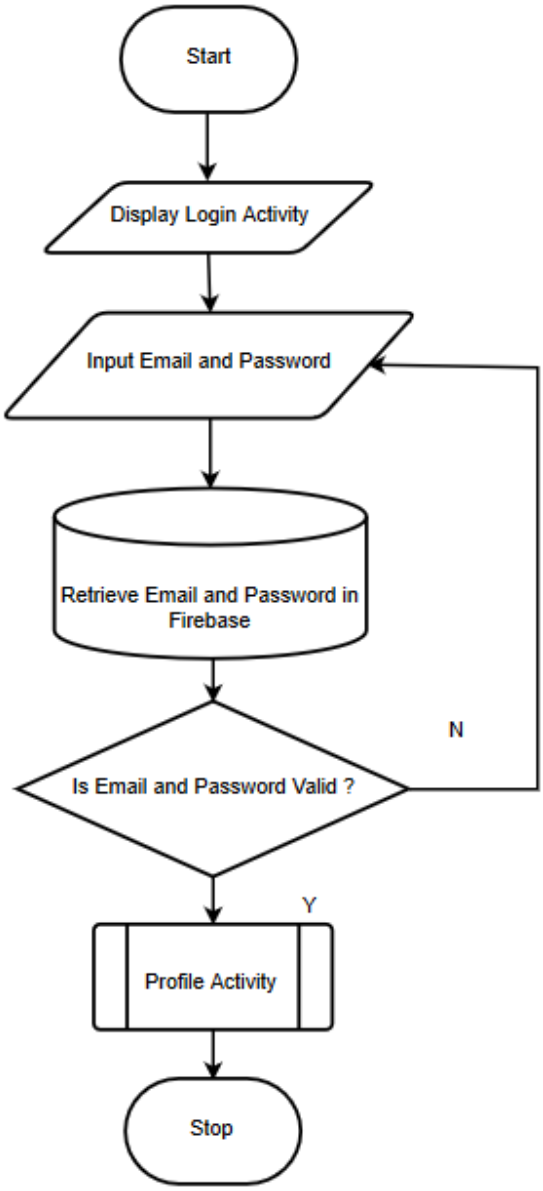

Figure 6. Login Activity

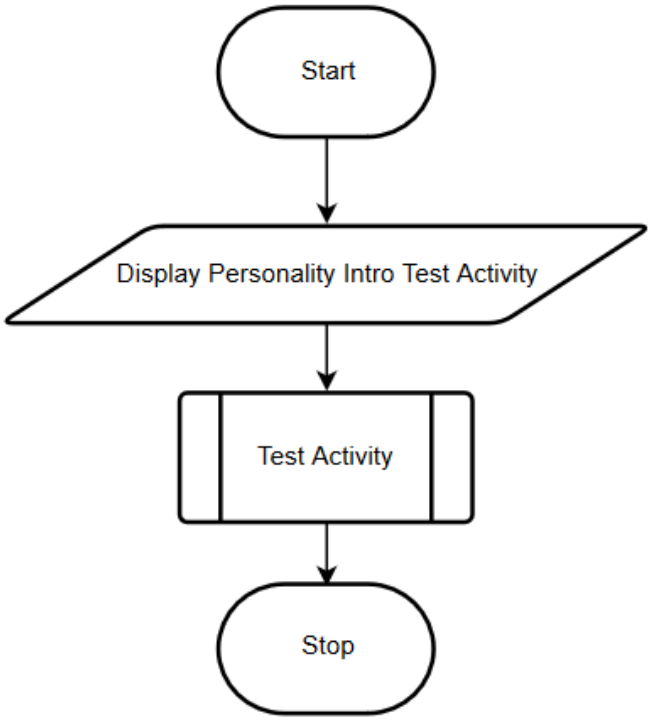

Figure 7. Personality Test Activity

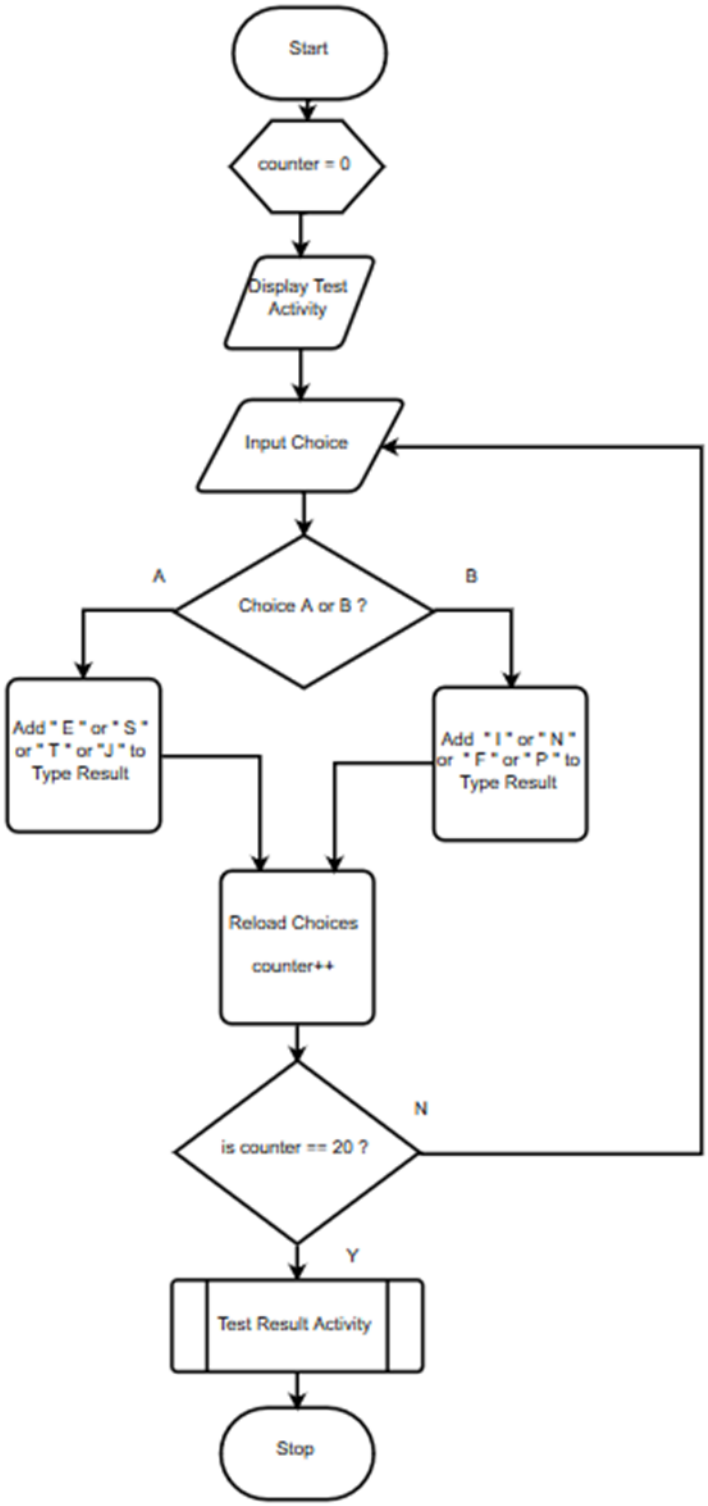

Figure 8. Test Activity/ Decision Tree Algorithm

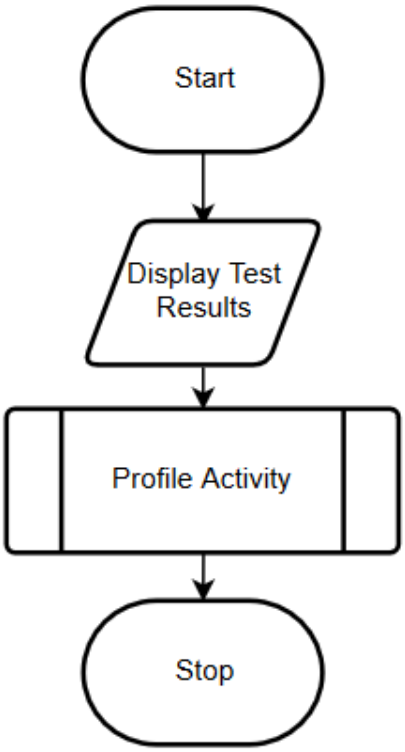

Figure 9. Test Result Activity

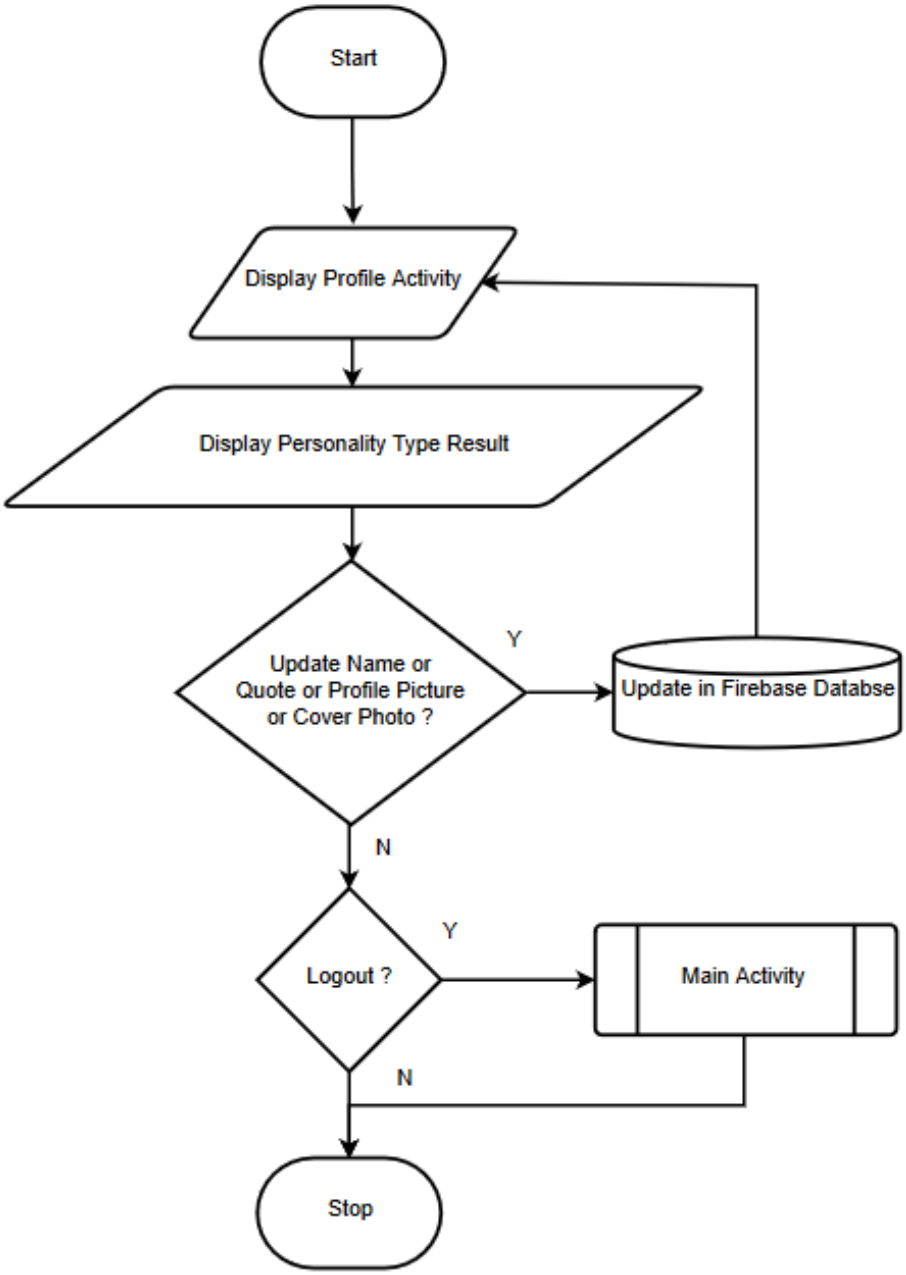

Figure 10. Profile Activity

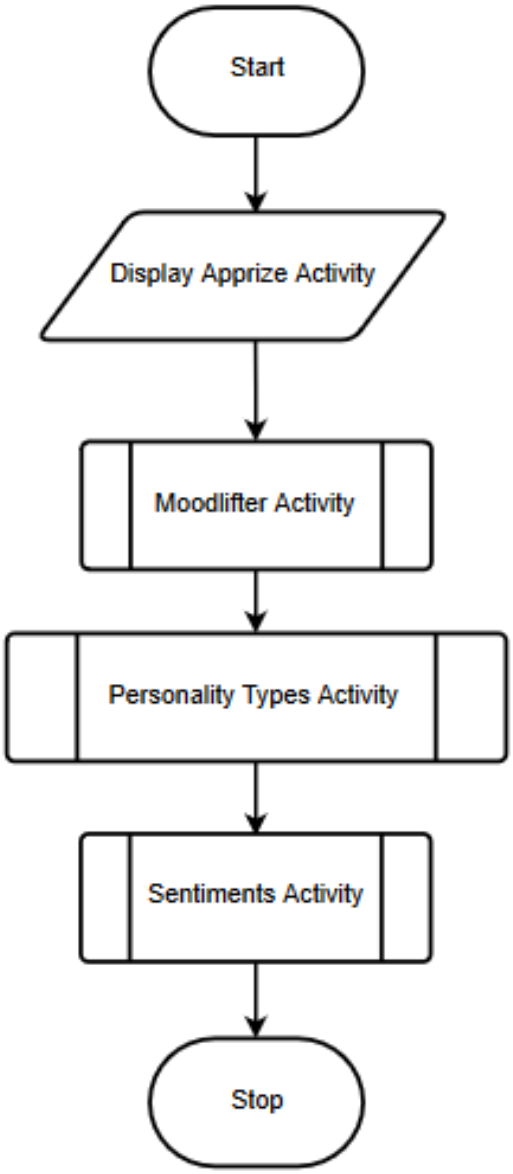

Figure 11. Apprize Activity

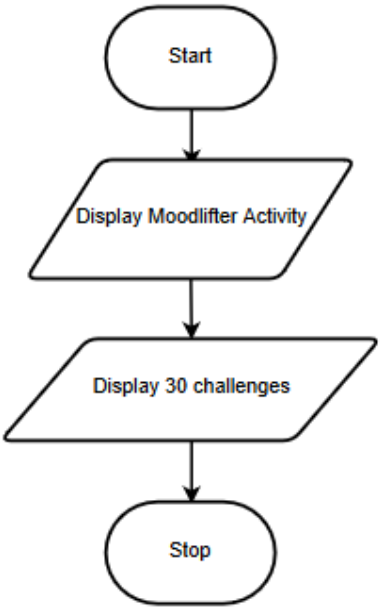

Figure 12. Mood Lifter Activity

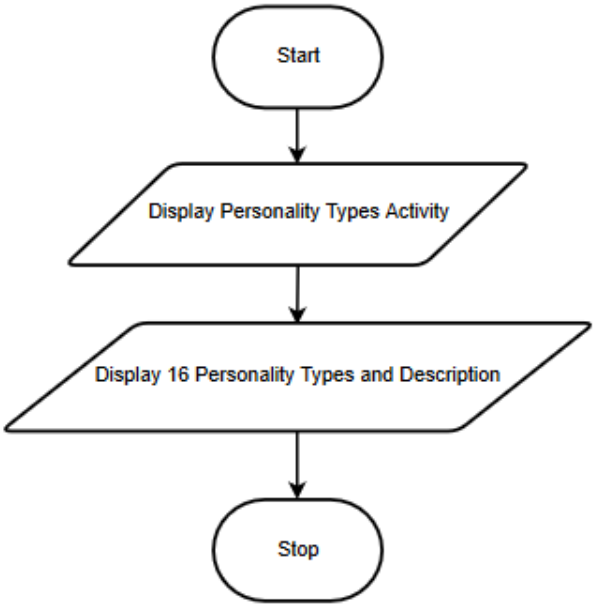

Figure 13. Personality Types Activity

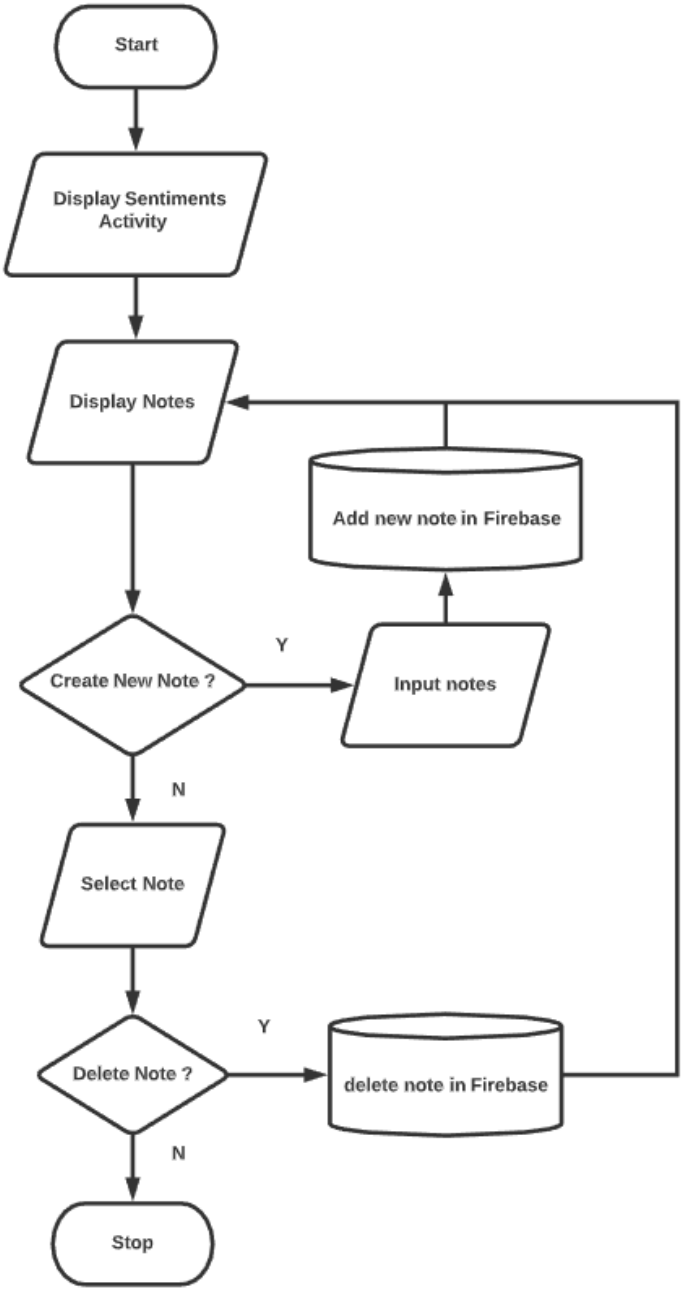

Figure 14. Sentiments Activity

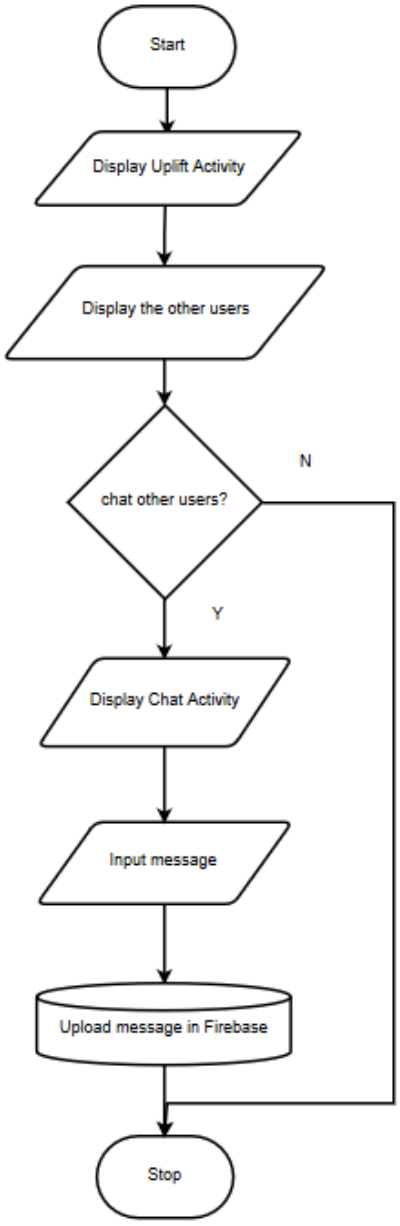

Figure 15. Uplift Activity

4.0 RESULTS AND DISCUSSIONS

Screen Layouts

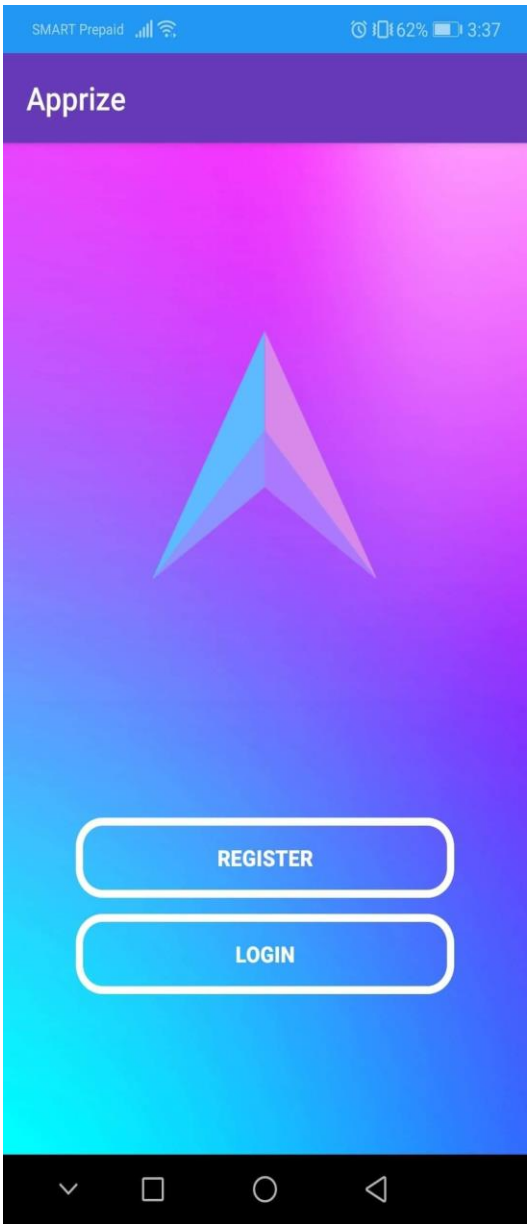

Figure 16. Main Activity

The main activity screen displays the start-up procedures of Apprize. The end-user is prompted to register to access the mobile application or login for existing data accounts.

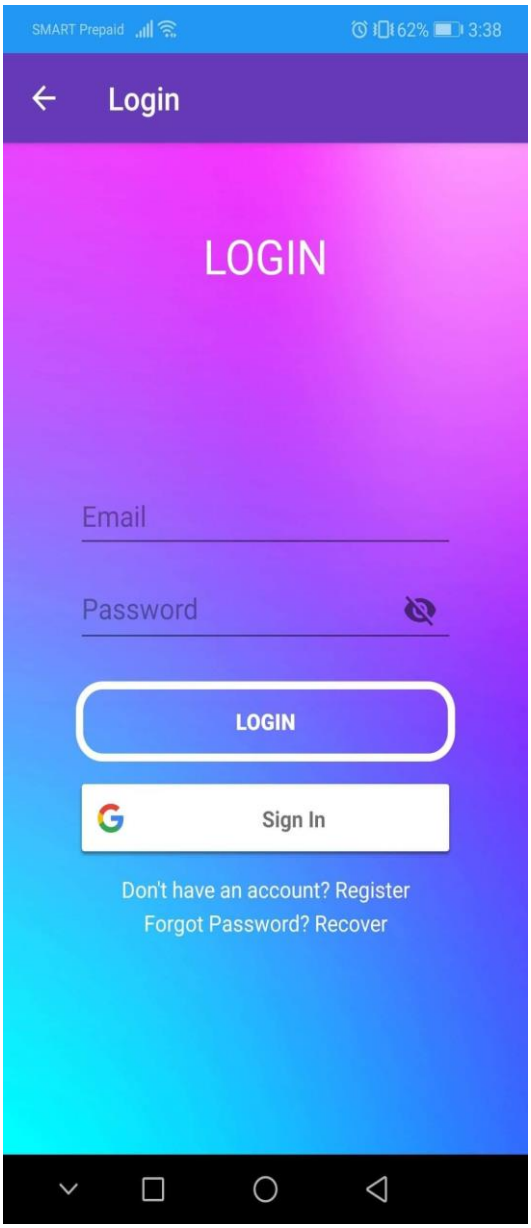

Figure 17. User Login

The user login screen displays the login procedure with the request of entering both the user name and password to access an account and use the mobile application.

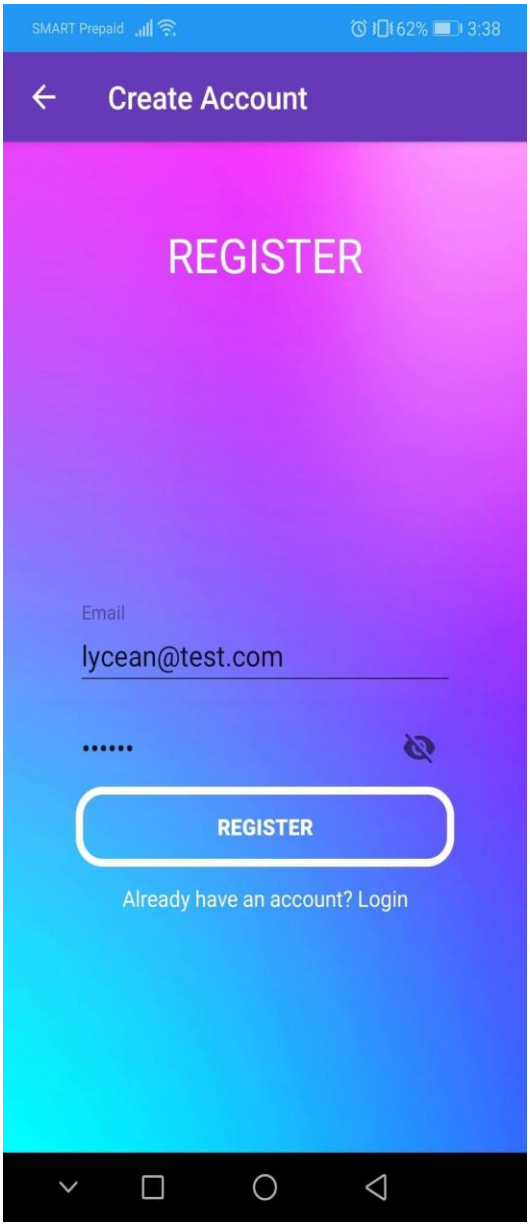

Figure 18. User Registration

The user registration screen presents the registration process with the request of inputting a new user name and password to be itemized in Google Firebase, for the end-user to utilize the mobile application.

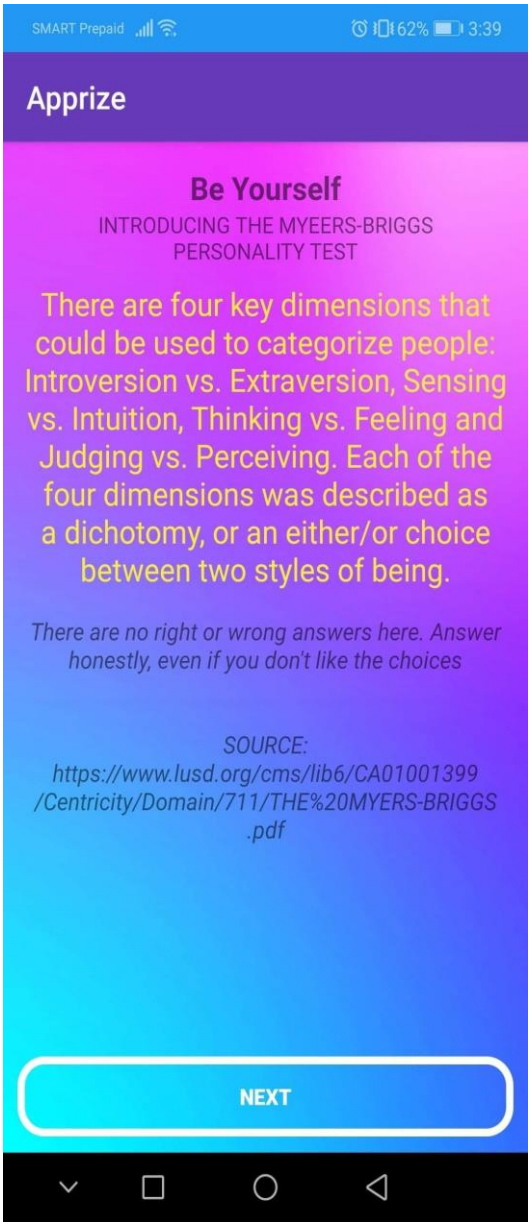

Figure 19. Personality Intro Test Activity

This respective screen displays the introduction of the mobile application's personality test. It explains what kind of examination the end-user will take and the possible results the test will output.

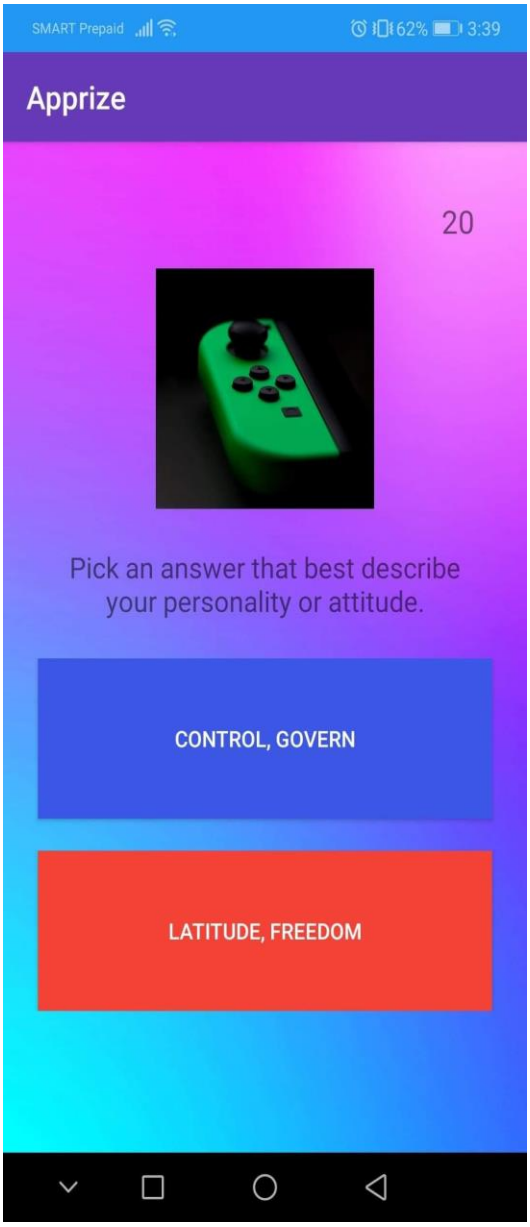

Figure 20. Test Activity

Figure 21. Test Results

The test activity screen provides 20 scenarios or set of choices where the end-user is encouraged to answer honestly. This is the part of the mobile application where the Decision Tree Algorithm was applied.

After the end-user has answered the personality test, the mobile application will display the end-user's initial personality trait based on the set of choices the user selected. The result includes a designation and description of the personality trait.

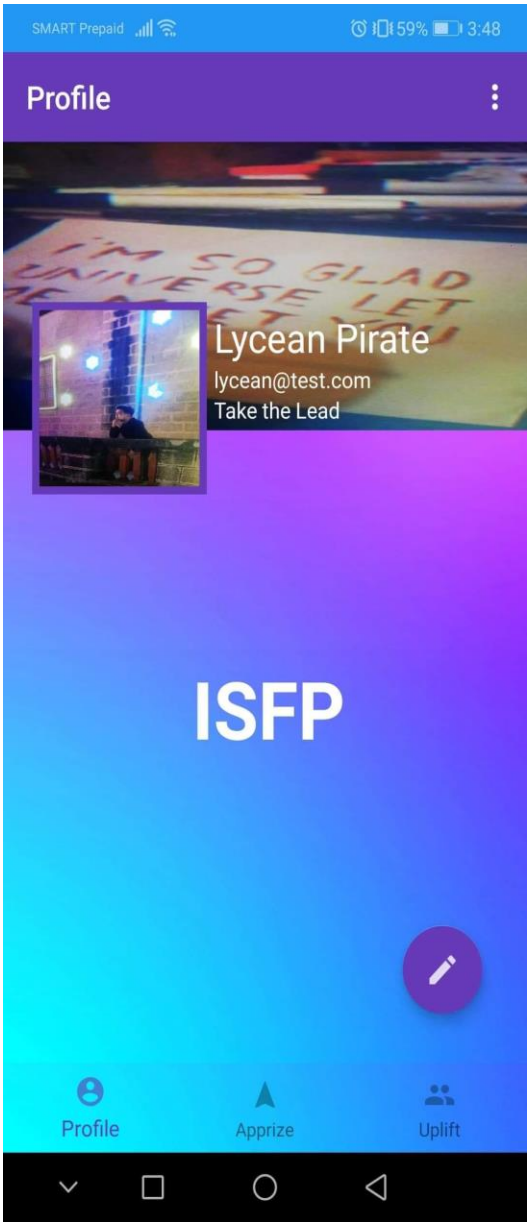

Figure 22. Home/Profile

The main menu which consists of the user Profile along with its tabs: Profile fragment, Apprize fragment, and the Uplift fragment. The profile displays the end-user's information along with the personality trait.

Figure 23. Apprize Activity

If the activity arrow button is pressed, it will be exhibited; displaying a variety of menus: Mood lifter, Personality Types, and Sentiments which will give the user a series of Apprize's functionalities to choose from.

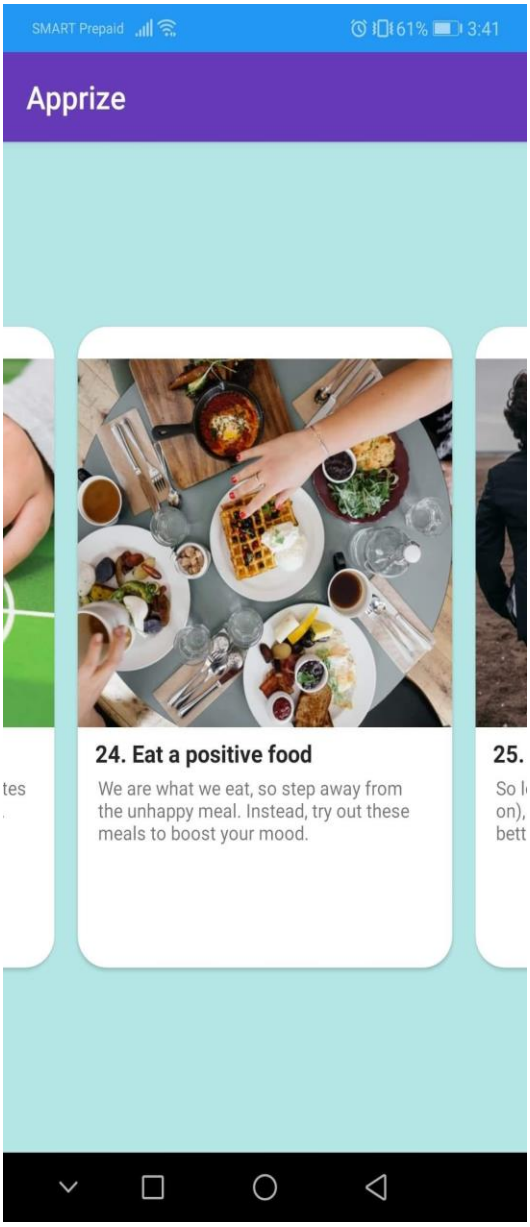

Figure 24. Mood Lifter

The Mood lifter option in the Apprize arrow tab unveils a variety of 30 simple tasks that is anticipated to boosts good vibes.

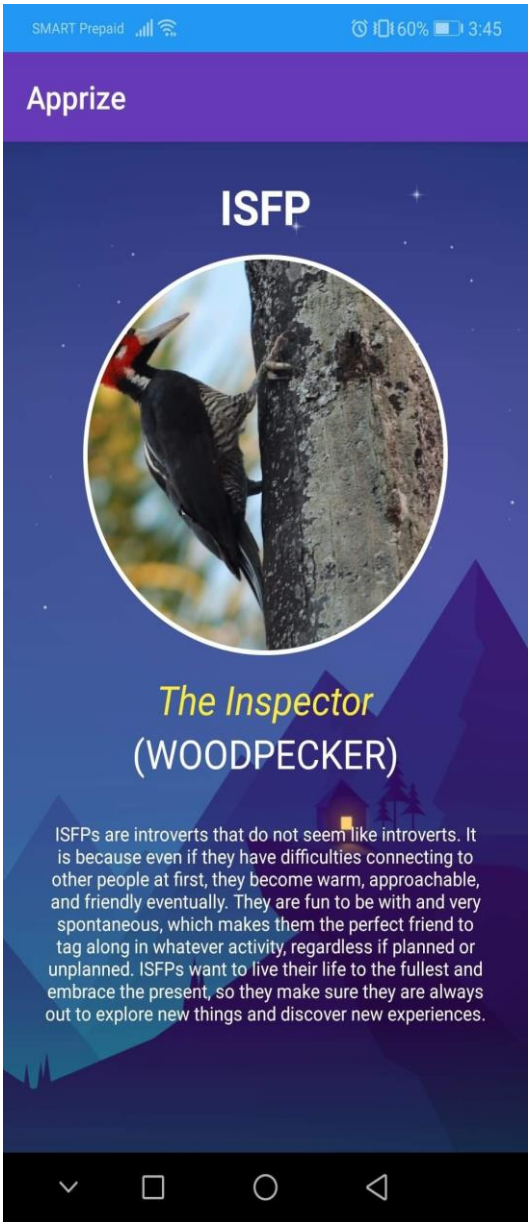

Figure 25. Personality Types

This screen displays the 16 personality types based on the Myers-Briggs Personality Test. It includes the designation and description of the personality trait alongside with its corresponding bird emblem or representation.

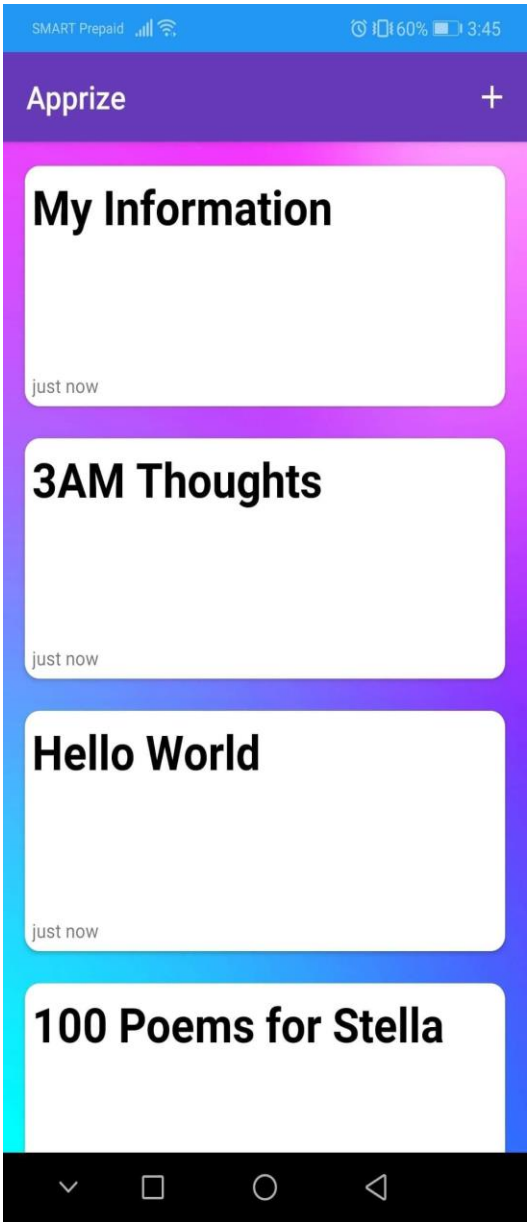

Figure 26. Sentiments/Journal

The Sentiments option in the Apprize tab if picked will be an outlet for end-users to express and input what they feel in a private manner in order to release excess thoughts.

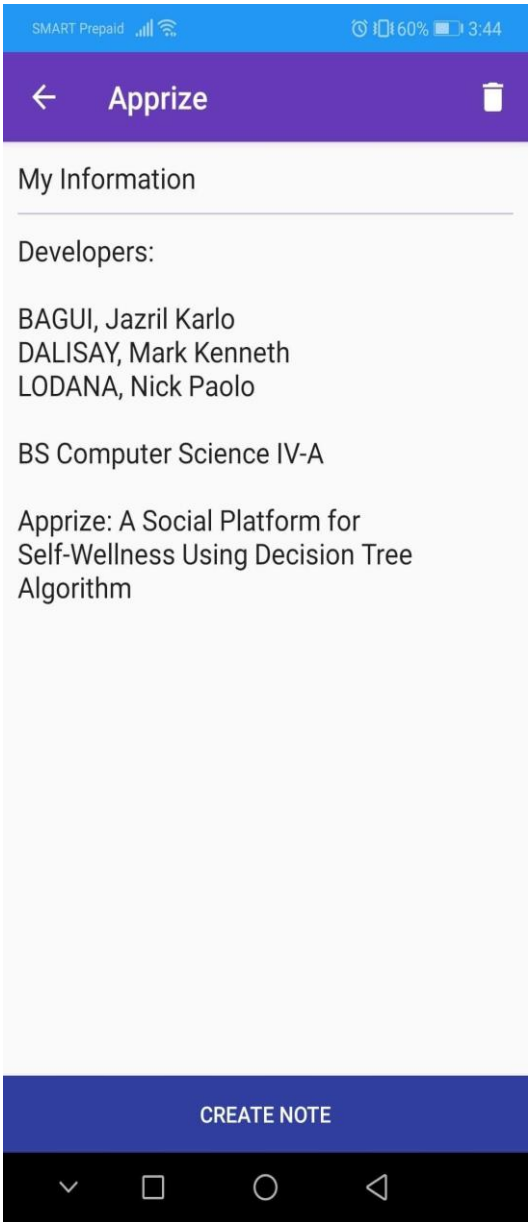

Figure 27. Create Note

The Create Note screen displays the function of the Sentiments fragment. It prompts the user to enter the note's title and content. It also allows the end-user to delete or edit the respective note or sentiment.

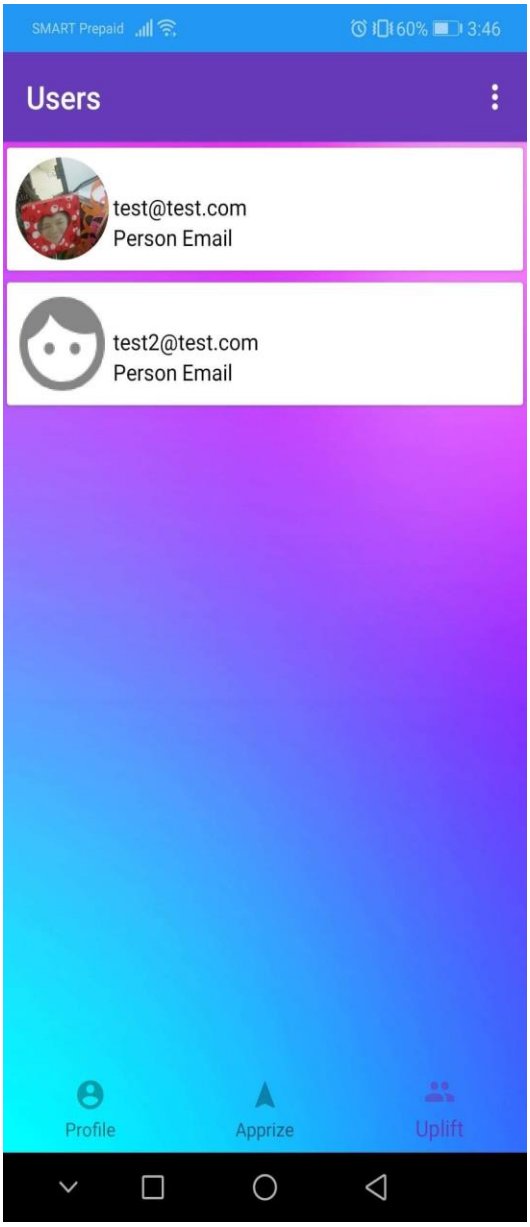

Figure 28. Uplift Activity

The Uplift fragment screen displays the mobile application's social platform where end-user will see the current users of Apprize for possible socialization and communication interest.

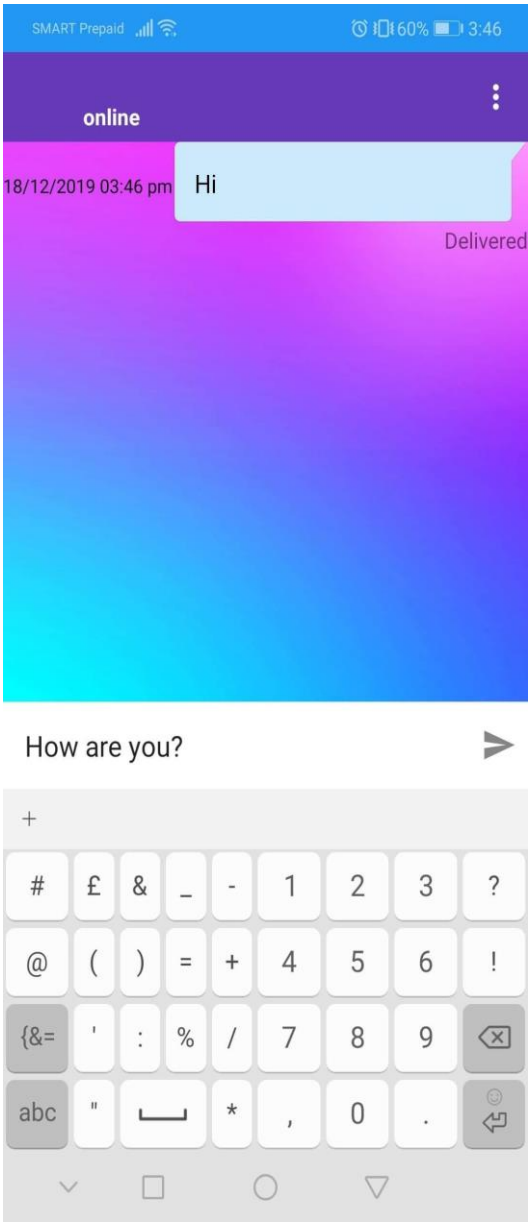

Figure 29. Chat Activity

When the end-user has selected a desired friend or fellow Apprize user to chat with, the chat activity will display the chat screen where the socialization process transpires.

The researchers tested the mobile application for its accuracy to handle the data provided. The choices “expend energy, enjoy groups " and " conserve energy, enjoy one-on-one ", " more outgoing, think out loud ", " more reserved, think to yourself ", " external, communicative, express yourself ", " internal, reticent, keep to yourself ", yielded the desired results needed for the accomplishment of the personality test. The results are based on the probability of which choice would the user selects to each set of options. The choices change every time the user choose an answer. The data that has been stored in a java class file is managed properly in each set of choices. The other cache for the data needed are stored in firebase. The data sent and retrieved are satisfyingly good depending on the internet connection the user has. The researchers created an account in the mobile application and uploaded a picture and a text, then the researchers retrieve that same data and displayed it on the profile activity of the mobile application.

5.0 CONCLUSIONS AND RECOMMENDATIONS

5.1 CONCLUSIONS

Based on the aforementioned features findings of the study, the following conclusions are hereby forwarded:

1. Technology can also be a powerful and efficient medium in mitigating the problem brought by loneliness, isolation and other mental health issues.
2. Cloud Storage and computing is a promising technological advancement that has an immense potential in dealing a positive effect in the digital world. It also provides many benefits and convenience to end-users.
3. Unstable internet connection affects the ability of the mobile application to perform its major features and functions.

5.2 RECOMMENDATIONS

Apprize is recommended for Android phone end-users. End-users must have a secure internet connection to utilize the functionalities and features of the mobile application. Apprize is a platform to improve an individual’s self-esteem and partially understand oneself through personality tests and connecting with other people.

Future potential researchers who may have an identical mindset towards diminishing the global widespread of loneliness or even advocating mental health awareness in general, are allowed to upgrade this thesis project. Since the Apprize was developed using a Cross-Platform Mobile Development Software, Android Studio in particular, future researchers may find ease in converting and deploying the mobile application in Windows and iOS smart phones. Researchers can also enhance the project by putting in an additional fragment to store and display from daily to the yearly results in personality or mood progresses in order for end-users to have a development overview. Other augmentations are significantly recommended, such as group chats, layout themes, access to professional help, create posts, in-app notifications and enriching user experience to completely utilize the mobile application.

REFERENCES

[1] Cacioppo, J.T. (2009) Journal of Personality and Social Psychology https://bit.ly/2qKJivV

[2] Personal Diversity in the Work Place https://bit.ly/2XVicNA

[3] Bires, Joseph. Diversity in the Workplace, https://bit.ly/2XVicNA

[4] FriendShare: An Algorithm of Finding Friends in a Social Networking Site, Juan Carlo P. Alvarez, Marivie M. Canovas, Jan

[5] MentalHelp.Net Why is Self-Esteem Important? https://bit.ly/2GRdKZk

[6] The Association Between Personality and Loneliness https://bit.ly/2RLblXl

[7] Truity. (2017). The 16 Personality TypeProfiles,https://www.truity.com/view/types

[8] Personality and Self-Esteem as Predictors of Young People's Technology Use https://bit.ly/36q21Mo

[9] Techtarget. (2019. Mobile App , https://bit.ly/2Ppj9K8

[10] https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb

[11] Agafonkin, Vladimir. (2017, April 27). A Dive Into Spatial Search Algorithms: Searching Through Millions in an Instant, https://blog.mapbox.com/a-dive-into-spatial-search-algorithms-ebd0c5e39d2a

[12] https://firebase.google.com

[13] Elena. (2017, February 16). 10 Apps that Promote Positivity, https://bit.ly/2UWPTMW

[14] Psyberguide. What's Up. 2018, https://bit.ly/2Hz2OkN

[15] Insight Timer. (2019, May 14.) https://bit.ly/2HRTfMR

[14] Talkspace.(2019) https://bit.ly/2YKgTla

[16] Code Blue: An App to Help Teenagers Experiencing Depression https://bit.ly/2JBQbHG

[17] ISO 9126 Software Quality Characteristics https://bit.ly/2sjKFCg

CODES LISTINGS

Main Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

  //views
  Button mLoginBtn,mRegisterBtn;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    //init views
    mLoginBtn = findViewById(R.id.login_btn);
    mRegisterBtn = findViewById(R.id.register_btn);

    //handle register button click
    mRegisterBtn.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {

        //Start register activity
        startActivity(new Intent(MainActivity.this,
RegisterActivity.class));
        finish();
      }
    });

    //handle login button click
    mLoginBtn.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {

        startActivity(new Intent(MainActivity.this,
LoginActivity.class));
        finish();
      }
    });


  }
}
```

Register Activity

```
package com.example.apprize;
import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import java.util.HashMap;

public class RegisterActivity extends AppCompatActivity {

  //views
  EditText mEmailEt, mPasswordEt;
  Button mRegisterBtn;
  TextView mHaveAccountTv;

  //progressbar
  ProgressDialog progressDialog;

  //Declare an instance of FirebaseAuth
  private FirebaseAuth mAuth;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);

    //Action bar and its title
    ActionBar actionBar = getSupportActionBar();
    actionBar.setTitle("Create Account");

    //enable back button
    actionBar.setDisplayHomeAsUpEnabled(true);
    actionBar.setDisplayShowHomeEnabled(true);

    //init
    mEmailEt = findViewById(R.id.emailEt);
    mPasswordEt = findViewById(R.id.passwordEt);
    mRegisterBtn = findViewById(R.id.registerBtn);
    mHaveAccountTv = findViewById(R.id.have_accountTv);

    progressDialog = new ProgressDialog(this);
    progressDialog.setMessage("Registering User...");

    //In the onCreate() method, initialize the FirebaseAuth
instance.
    mAuth = FirebaseAuth.getInstance();

    //handle have account text click
    mHaveAccountTv.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {
```

```
            //Go to Login Activity
            startActivity(new Intent(RegisterActivity.this,
LoginActivity.class));
            finish();
        }
    });

    //handle register button click
    mRegisterBtn.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

            //input email and password
            String email = mEmailEt.getText().toString().trim();
            String password =
mPasswordEt.getText().toString().trim();

            //validate
            if
(!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
                //set error focus to email edit text
                mEmailEt.setError("Invalid Email");
                mEmailEt.setFocusable(true);
            }
            else if(password.length() < 6) {
                //set error focus to password edit text
                mPasswordEt.setError("Password length at least
6 characters");
                mPasswordEt.setFocusable(true);
            }
            else {
                //register the user
                registerUser(email,password);
            }
        }
    });
  }

  private void registerUser(String email, String password) {

    //email and password valid show progress dialog
    progressDialog.show();

    mAuth.createUserWithEmailAndPassword(email,
password)
            .addOnCompleteListener(RegisterActivity.this, new
OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull
Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        // Sign in success, dismiss dialog and start
activity
                        progressDialog.dismiss();
                        FirebaseUser user = mAuth.getCurrentUser();

                        //get user email and uid from auth
                        String email = user.getEmail();
                        String uid = user.getUid();

                        //When user is registered store user info in
firebase realtime database too
                        //using hashmap
                        HashMap<Object, String> hashMap = new
HashMap<>();
                        //path to store user data named "users
                        //put info in hashmap
                        hashMap.put("email", email);
                        hashMap.put("uid", uid);
                        hashMap.put("name", "");
                        hashMap.put("quote", "");
                        hashMap.put("image", "");
                        hashMap.put("cover", "");
                        //firebase database instance
                        FirebaseDatabase database =
FirebaseDatabase.getInstance();
                        //path to store user data named "Users"
                        DatabaseReference reference =
database.getReference("Users");
                        //put data within hashmap in database
                        reference.child(uid).setValue(hashMap);

                        Toast.makeText(RegisterActivity.this,
"Registered...\n"+user.getEmail(),
Toast.LENGTH_SHORT).show();
                        startActivity(new Intent(RegisterActivity.this,
PersonalityIntroActivity.class));
                        finish();

                    } else {
                        // If sign in fails, display a message to the
user.
                        progressDialog.dismiss();
                        Toast.makeText(RegisterActivity.this,
"Authentication failed.",
                                Toast.LENGTH_SHORT).show();

                    }


                }
            }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            //error, dismiss progress dialog and get message
            progressDialog.dismiss();
            Toast.makeText(RegisterActivity.this,
""+e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });

  }

  @Override
  public boolean onSupportNavigateUp() {
    onBackPressed();// go previous activity
    return super.onSupportNavigateUp();
  }
}
```

Login Activity

```
package com.example.apprize;
import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.InputType;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import
com.google.android.gms.auth.api.signin.GoogleSignIn;
import
com.google.android.gms.auth.api.signin.GoogleSignInAccou
nt;
import
com.google.android.gms.auth.api.signin.GoogleSignInClient;
import
com.google.android.gms.auth.api.signin.GoogleSignInOption
s;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import java.util.HashMap;

public class LoginActivity extends AppCompatActivity {

   private static final int RC_SIGN_IN = 100 ;
   GoogleSignInClient mGoogleSignInClient;

   //views
   EditText mEmailEt, mPasswordEt;
   TextView mNotHaveAccount, mRecoverPassTv;
   Button mLoginBtn;
   SignInButton mGoogleLoginBtn;

   //Declare an instance of FirebaseAuth
   private FirebaseAuth mAuth;

   //progressbar
   ProgressDialog pd;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_login);

      //Action bar and its title
      ActionBar actionBar = getSupportActionBar();
      actionBar.setTitle("Login");

      //enable back button
      actionBar.setDisplayHomeAsUpEnabled(true);
      actionBar.setDisplayShowHomeEnabled(true);

      //before mAuth
      // Configure Google Sign In
      GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT
_SIGN_IN)

.requestIdToken(getString(R.string.default_web_client_id))
            .requestEmail()
            .build();
      mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

      //In the onCreate() method, initialize the FirebaseAuth
instance.
      mAuth = FirebaseAuth.getInstance();

      //init views
      mEmailEt = findViewById(R.id.emailEt);
      mPasswordEt = findViewById(R.id.passwordEt);
      mNotHaveAccount =
findViewById(R.id.nothave_accountTv);
      mRecoverPassTv = findViewById(R.id.recoverPassTv);
      mLoginBtn = findViewById(R.id.loginBtn);
      mGoogleLoginBtn = findViewById(R.id.googleLoginBtn);

      //handle login button click
      mLoginBtn.setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View view) {

            //input data
            String email = mEmailEt.getText().toString();
            String passw =
mPasswordEt.getText().toString().trim();
            if
(!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
               //invalid email pattern set error
               mEmailEt.setError("Invalid Email");
               mEmailEt.setFocusable(true);
            }
            else {
               //valid email pattern
               loginUser(email,passw);
            }
         }
      });

      //handle not have account button click
      mNotHaveAccount.setOnClickListener(new
View.OnClickListener() {
```

```
        @Override
        public void onClick(View view) {

            startActivity(new Intent(LoginActivity.this,
RegisterActivity.class));
            finish();
        }
    });

    //recover pass text click
    mRecoverPassTv.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

            showRecoverPasswordDialog();

        }
    });

    //handle google login btn click
    mGoogleLoginBtn.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //begin google login process
            Intent signInIntent =
mGoogleSignInClient.getSignInIntent();
            startActivityForResult(signInIntent, RC_SIGN_IN);
        }
    });

    //init progress dialog
    pd = new ProgressDialog(this);
  }

  private void showRecoverPasswordDialog() {

    //Alert Dialog
    AlertDialog.Builder builder = new
AlertDialog.Builder(this);
    builder.setTitle("Recover Password");

    //set layout linear layout
    LinearLayout linearLayout = new LinearLayout(this);

    //views to set to dialog
    final EditText emailEt = new EditText(this);
    emailEt.setHint("Email");

emailEt.setInputType(InputType.TYPE_TEXT_VARIATION_EM
AIL_ADDRESS);
    emailEt.setMinEms(16);
    linearLayout.addView(emailEt);
    linearLayout.setPadding(10,10,10,10);
    builder.setView(linearLayout);

    //buttons recover
    builder.setPositiveButton("Recover", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int
i) {

            //inputemail
            String email = emailEt.getText().toString().trim();
            beginRecovery(email);
        }
    });
    //buttons cancel
    builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int
i) {

            //dismiss dialog
            dialogInterface.dismiss();

        }
    });

    //show dialog
    builder.create().show();

  }

  private void beginRecovery(String email) {
    //show progress dialog
    pd.setMessage("Sending email...");

    pd.show();


mAuth.sendPasswordResetEmail(email).addOnCompleteList
ener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
            pd.dismiss();
            if (task.isSuccessful()) {
                Toast.makeText(LoginActivity.this, "Email sent",
Toast.LENGTH_SHORT).show();
            }
            else {
                Toast.makeText(LoginActivity.this, "Failed",
Toast.LENGTH_SHORT).show();
            }
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            pd.dismiss();
            //get and show proper error message
            Toast.makeText(LoginActivity.this,
""+e.getMessage(), Toast.LENGTH_SHORT).show();
        }
    });
  }

  private void loginUser(String email, String passw) {

    //show progress dialog
```

```
        pd.setMessage("Logging In...");
        pd.show();
        mAuth.signInWithEmailAndPassword(email, passw)
                .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull
Task<AuthResult> task) {
                        if (task.isSuccessful()) {

                            // Sign in success, update UI with the signed-
in user's information
                            FirebaseUser user = mAuth.getCurrentUser();
                                pd.dismiss();
                            startActivity(new Intent(LoginActivity.this,
DashboardActivity.class));
                            finish();

                        } else {

                            // If sign in fails, display a message to the
user.
                            Toast.makeText(LoginActivity.this,
"Authentication failed.",
                                    Toast.LENGTH_SHORT).show();
                            pd.dismiss();
                        }
                    }
                }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                pd.dismiss();
                Toast.makeText(LoginActivity.this,
""+e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
    }

    @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();// go previous activity
        return super.onSupportNavigateUp();
    }
    @Override
    public void onActivityResult(int requestCode, int
resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
        if (requestCode == RC_SIGN_IN) {
            Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
            try {
                // Google Sign In was successful, authenticate with
Firebase
                GoogleSignInAccount account =
task.getResult(ApiException.class);
                firebaseAuthWithGoogle(account);
            } catch (ApiException e) {
                // Google Sign In failed, update UI appropriately
                Toast.makeText(this,""+e.getMessage(),
Toast.LENGTH_SHORT).show();
            }
        }
    }
    private void
firebaseAuthWithGoogle(GoogleSignInAccount acct) {

        AuthCredential credential =
GoogleAuthProvider.getCredential(acct.getIdToken(), null);
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull
Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            // Sign in success, update UI with the signed-
in user's information
                            FirebaseUser user = mAuth.getCurrentUser();

                            //if user is signing in first time then get and
show user info from google account
                            if
(task.getResult().getAdditionalUserInfo().isNewUser()) {
                                //get user email and uid from auth
                                String email = user.getEmail();
                                String uid = user.getUid();
                                //When user is registered store user info in
firebase realtime database too
                                //using hashmap
                                HashMap<Object, String> hashMap = new
HashMap<>();
                                //path to store user data named "users
                                //put info in hashmap
                                hashMap.put("email", email);
                                hashMap.put("uid", uid);
                                hashMap.put("name", "");
                                hashMap.put("quote", "");
                                hashMap.put("image", "");
                                hashMap.put("cover", "");
                                //firebase database instance
                                FirebaseDatabase database =
FirebaseDatabase.getInstance();
                                //path to store user data named "Users"
                                DatabaseReference reference =
database.getReference("Users");
                                //put data within hashmap in database
                                reference.child(uid).setValue(hashMap);

                                //startActivity(new
Intent(LoginActivity.this, PersonalityIntroActivity.class));
                                //finish();

                            }
                            //show user email in toast
                            Toast.makeText(LoginActivity.this,
""+user.getEmail(), Toast.LENGTH_SHORT).show();
                            //go to profile activity after logged in
                            startActivity(new Intent(LoginActivity.this,
PersonalityIntroActivity.class));
```

```
                    finish();
                    //updateUI(user);
                } else {
                    // If sign in fails, display a message to the
user.
                    Toast.makeText(LoginActivity.this, "Login
Failed...", Toast.LENGTH_SHORT).show();
                    //updateUI(null);
                }
            }
        }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            //get and show error message
            Toast.makeText(LoginActivity.this,
""+e.getMessage(), Toast.LENGTH_SHORT).show();

        }
    });
  }

}
```

Personality Test Intro Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import com.google.firebase.database.DatabaseReference;

public class PersonalityIntroActivity extends
AppCompatActivity {

  Button mStartTestBtn;

  @Override
  protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_personality_intro);
    mStartTestBtn = findViewById(R.id.start_testBtn);
    mStartTestBtn.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {

        startActivity(new
Intent(PersonalityIntroActivity.this, EITestActivity.class));
      }
    });
  }
}
```

Extravert/Introvert (E/I) Test Activity

```
package com.example.apprize;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import com.google.firebase.database.FirebaseDatabase;
import java.util.ArrayList;

public class EITestActivity extends AppCompatActivity {

  private TextView mQuestionNumberView, mQuestion;
  private ImageView mImageView;
  private Button mTrueButton, mFalseButton;

  int ecounter;
  int icounter;

  private int m = 1;
  private int mQuestionNumber = 0;

  @Override
  protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_eitest);
    final ArrayList<String> alist = new ArrayList<>();
    FirebaseDatabase database =
FirebaseDatabase.getInstance();
    mQuestionNumberView = findViewById(R.id.points);
    mImageView = findViewById(R.id.imageView);
    mQuestion = findViewById(R.id.question);
    mTrueButton = findViewById(R.id.aButton);
    mFalseButton = findViewById(R.id.bButton);
    updateQuestion();
    updateType();

    //Logic for true button
    mTrueButton.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {

        if(mQuestionNumber == 0 || mQuestionNumber ==
1 || mQuestionNumber == 2 || mQuestionNumber == 3 ||
mQuestionNumber == 4) {
          updateType();
          m++;
          ecounter++;
          alist.add("E");

          //perform check before you update the question
          if (mQuestionNumber ==
EIBook.questions.length) {
            Intent intent = new Intent(EITestActivity.this,
SNTestActivity.class);
            intent.putExtra("finalType1", alist);
            startActivity(intent);
```

```
            } else {
                updateQuestion();
                updateType();
            }
            result();
        }
        else {

            //perform check before you update the question
            if (mQuestionNumber ==
EIBook.questions.length) {
                Intent intent = new Intent(EITestActivity.this,
SNTestActivity.class);
                intent.putExtra("finalType1", alist);
                startActivity(intent);
            } else {
                updateQuestion();
                updateType();
            }
        }
    }
});

    //Logic for false button
    mFalseButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

            if(mQuestionNumber == 0 || mQuestionNumber ==
1 || mQuestionNumber == 2 || mQuestionNumber == 3 ||
mQuestionNumber == 4) {
                updateType();
                m++;
                icounter++;
                alist.add("I");

                //perform check before you update the question
                if (mQuestionNumber ==
EIBook.questions.length) {
                    Intent intent = new Intent(EITestActivity.this,
SNTestActivity.class);
                    intent.putExtra("finalType1", alist);
                    startActivity(intent);
                } else {
                    updateQuestion();
                    updateType();
                }
                result();
            }
            else {

                //perform check before you update the question
                if (mQuestionNumber ==
EIBook.questions.length) {
                    Intent intent = new Intent(EITestActivity.this,
SNTestActivity.class);
                    intent.putExtra("finalType1", alist);
                    startActivity(intent);
                } else {
                    updateQuestion();
                    updateType();
                }
            }

        }
    });

}

private void updateType() {

    mQuestionNumberView.setText("" + m);

}

private void updateQuestion() {


mImageView.setImageResource(EIBook.images[mQuestionN
umber]);

mTrueButton.setText(EIBook.getChoice1(mQuestionNumber
));

mFalseButton.setText(EIBook.getChoice2(mQuestionNumbe
r));

mQuestion.setText(EIBook.questions[mQuestionNumber]);
    mQuestionNumber++;
}

public void result (){

    if (m >= 5) {

        SharedPreferences pref;
        SharedPreferences.Editor editor;
        pref = getSharedPreferences("Apprize",
MODE_PRIVATE);
        editor = pref.edit();

        if (ecounter > icounter) {
            editor.putString("mEItype1", "E");
        } else {
            editor.putString("mEItype1", "I");
        }
        editor.apply();
    }
}
}
```

Extravert/Introvert (E/I) Book.java

```
package com.example.apprize;
public class EIBook {

    public static String mChoices[][] = {
            {" expend energy, enjoy groups ", " conserve
energy, enjoy one-on-one "},
            {" more outgoing, think out loud ", " more reserved,
think to yourself "},
```

```
            {" seek many tasks, public activities, interaction
with others ", " seek private, solitary activities with quiet to
concentrate "},
            {" external, communicative, express yourself ", "
internal, reticent, keep to yourself "},
            {" active, initiate ", " reflective, deliberate "},
        };

        public static String[] questions = new String [] {
            "Pick an answer that best describe your personality
or attitude.",
            "Pick an answer that best describe your personality
or attitude.",
            "Pick an answer that best describe your personality
or attitude.",
            "Pick an answer that best describe your personality
or attitude.",
            "Pick an answer that best describe your personality
or attitude."

        };

        public static String getChoice1(int a) {
            String choice0 = mChoices[a][0];
            return choice0;
        }

        public static String getChoice2(int a) {
            String choice1 = mChoices[a][1];
            return choice1;
        }

        public static int[] images = new int [] {
            R.drawable.energi, R.drawable.think,
R.drawable.seek, R.drawable.express,
            R.drawable.active
        };

    public static String[] description = new String [] {
            "Extraverts are energized by social gatherings, parties
and group activities. Extraverts are usually enthusiastic,
gregarious and animated. Their communication style is
verbal and assertive. Talking helps Extraverts think. They
enjoy limelight.",
            "Introverts are energized by spending time alone or
with a small group. They find large group gatherings draining
because they seek depth instead of breadth of relationships.
Introverts process information internally. They are great
listeners and think before talking."
    };
}
```

Sensing/Intuition (S/N) Test Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import com.google.firebase.database.FirebaseDatabase;
import java.util.ArrayList;

public class SNTestActivity extends AppCompatActivity {

    private TextView mQuestionNumberView, mQuestion;
    private ImageView mImageView;
    private Button mTrueButton, mFalseButton;
    int scounter;
    int ncounter;
    private int m = 6;
    private int mQuestionNumber = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sntest);
        final ArrayList<String> blist = new ArrayList<>();
        FirebaseDatabase database =
FirebaseDatabase.getInstance();
        mQuestionNumberView = findViewById(R.id.points);
        mImageView = findViewById(R.id.imageView);
        mQuestion = findViewById(R.id.question);
        mTrueButton = findViewById(R.id.aButton);
        mFalseButton = findViewById(R.id.bButton);
        updateQuestion();
        updateType();

        //Logic for true button
        mTrueButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if (mQuestionNumber == 0 || mQuestionNumber
== 1 || mQuestionNumber == 2 || mQuestionNumber == 3
|| mQuestionNumber == 4) {
                    updateType();
                    m++;
                    blist.add("S");

                    //perform check before you update the question
                    if (mQuestionNumber ==
SNBook.questions.length) {
                        Intent intent = new Intent(SNTestActivity.this,
TFTestActivity.class);
                        intent.putExtra("finalType2", blist);
                        startActivity(intent);
                    } else {
                        updateQuestion();
                        updateType();
                    }
                    result();
                } else {

                    //perform check before you update the question
                    if (mQuestionNumber ==
SNBook.questions.length) {
```

```
                Intent intent = new Intent(SNTestActivity.this,
TFTestActivity.class);
                intent.putExtra("finalType2", blist);
                startActivity(intent);
              } else {
                updateQuestion();
                updateType();
              }
            }
          }
      });

      //Logic for false button
      mFalseButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {

          if (mQuestionNumber == 0 || mQuestionNumber
== 1 || mQuestionNumber == 2 || mQuestionNumber == 3
|| mQuestionNumber == 4) {
            updateType();
            m++;
            blist.add("N");

            //perform check before you update the question
            if (mQuestionNumber ==
SNBook.questions.length) {
              Intent intent = new Intent(SNTestActivity.this,
TFTestActivity.class);
              intent.putExtra("finalType2", blist);
              startActivity(intent);
            } else {
              updateQuestion();
              updateType();
            }
            result();
          } else {

            //perform check before you update the question
            if (mQuestionNumber ==
SNBook.questions.length) {
              Intent intent = new Intent(SNTestActivity.this,
TFTestActivity.class);
              intent.putExtra("finalType2", blist);
              startActivity(intent);
            } else {
              updateQuestion();
              updateType();
            }
          }
        }
      });
  }

  private void updateType() {

    mQuestionNumberView.setText("" + m);
  }

  private void updateQuestion() {
mImageView.setImageResource(SNBook.images[mQuestion
Number]);

mTrueButton.setText(SNBook.getChoice1(mQuestionNumbe
r));

mFalseButton.setText(SNBook.getChoice2(mQuestionNumb
er));

mQuestion.setText(SNBook.questions[mQuestionNumber]);
    mQuestionNumber++;
  }

  public void result() {

    if (m >= 5) {
      SharedPreferences pref;
      SharedPreferences.Editor editor;
      pref = getSharedPreferences("Apprize",
MODE_PRIVATE);
      editor = pref.edit();
      if (scounter > ncounter) {
        editor.putString("mSNtype2", "S");
      } else
        editor.putString("mSNtype2", "N");
      editor.apply();
    }
    //return ret;
  }
}
```

Sensing/Intuition (S/N) Book.java

```
package com.example.apprize;
public class SNBook {

  public static String mChoices[][] = {
      {" interpret literally ", " look for meaning and
possibilities "},
      {" practical, realistic, experiential ", " imaginative,
innovative, theoretical "},
      {" standard, usual, conventional ", " different, novel,
unique "},
      {" focus on here-and-now ", " look to the future,
global perspective, “big picture” "},
      {" facts, things, “what is” ", " ideas, dreams, “what
could be,” philosophical "}
  };

  public static String[] questions = new String [] {
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude."
```

```
    };

    public static String getChoice1(int a) {
        String choice0 = mChoices[a][0];
        return choice0;
    }

    public static String getChoice2(int a) {
        String choice1 = mChoices[a][1];
        return choice1;
    }

    public static int[] images = new int [] {
            R.drawable.meaning, R.drawable.imagine,
R.drawable.different,
            R.drawable.focus, R.drawable.dreams
    };

    public static String[] description = new String [] {
            "Sensors focus on the present. They are “here and
now” people. They are factual and process information
through the five senses. They see things as they are because
they are concrete and literal thinkers. They trust what is
certain. Sensors value realism and common sense. They
especially like ideas with practical applications.",
            "iNtuitive people live in the future and are immersed
in the world of possibilities. They process information
through patterns and impressions. Intuitive people value
inspiration and imagination. They gather knowledge by
reading between the lines. Their abstract nature attracts
them toward deep ideas and concepts"
    };
}
```

Thinking/Feeling (T/F) Test Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import com.google.firebase.database.FirebaseDatabase;
import java.util.ArrayList;

public class TFTestActivity extends AppCompatActivity {

    private TextView mQuestionNumberView, mQuestion;
    private ImageView mImageView;
    private Button mTrueButton, mFalseButton;
    int tcounter;
    int fcounter;
    private int m = 11;
    private int mQuestionNumber = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_tftest);
        final ArrayList<String> clist = new ArrayList<>();
        FirebaseDatabase database =
FirebaseDatabase.getInstance();
        mQuestionNumberView = findViewById(R.id.points);
        mImageView = findViewById(R.id.imageView);
        mQuestion = findViewById(R.id.question);
        mTrueButton = findViewById(R.id.aButton);
        mFalseButton = findViewById(R.id.bButton);
        updateQuestion();
        updateType();

        //Logic for true button
        mTrueButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if (mQuestionNumber == 0 || mQuestionNumber
== 1 || mQuestionNumber == 2 || mQuestionNumber == 3
|| mQuestionNumber == 4) {
                    updateType();
                    m++;
                    clist.add("T");

                    //perform check before you update the question
                    if (mQuestionNumber ==
TFBook.questions.length) {
                        Intent intent = new Intent(TFTestActivity.this,
JPTestActivity.class);
                        intent.putExtra("finalType3", clist);
                        startActivity(intent);
                    } else {
                        updateQuestion();
                        updateType();
                    }
                    result();
                } else {

                    //perform check before you update the question
                    if (mQuestionNumber ==
TFBook.questions.length) {
                        Intent intent = new Intent(TFTestActivity.this,
JPTestActivity.class);
                        intent.putExtra("finalType3", clist);
                        startActivity(intent);
                    } else {
                        updateQuestion();
                        updateType();
                    }
                }
            }
        });

        //Logic for false button
        mFalseButton.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View view) {
```

```
            if (mQuestionNumber == 0 || mQuestionNumber
== 1 || mQuestionNumber == 2 || mQuestionNumber == 3
|| mQuestionNumber == 4) {
                updateType();
                m++;
                clist.add("F");

                //perform check before you update the question
                if (mQuestionNumber ==
TFBook.questions.length) {
                    Intent intent = new Intent(TFTestActivity.this,
JPTestActivity.class);
                    intent.putExtra("finalType3", clist);
                    startActivity(intent);
                } else {
                    updateQuestion();
                    updateType();
                }
                result();
            } else {

                //perform check before you update the question
                if (mQuestionNumber ==
TFBook.questions.length) {
                    Intent intent = new Intent(TFTestActivity.this,
JPTestActivity.class);
                    intent.putExtra("finalType3", clist);
                    startActivity(intent);
                } else {
                    updateQuestion();
                    updateType();
                }
            }
        }
    });
  }

  private void updateType() {

    mQuestionNumberView.setText("" + m);
  }

  private void updateQuestion() {


mImageView.setImageResource(TFBook.images[mQuestion
Number]);

mTrueButton.setText(TFBook.getChoice1(mQuestionNumbe
r));

mFalseButton.setText(TFBook.getChoice2(mQuestionNumbe
r));

mQuestion.setText(TFBook.questions[mQuestionNumber]);
    mQuestionNumber++;
  }

  public void result() {
    if (m >= 5) {
      SharedPreferences pref;
      SharedPreferences.Editor editor;
      pref = getSharedPreferences("Apprize",
MODE_PRIVATE);
      editor = pref.edit();

      if (tcounter > fcounter) {
        editor.putString("mTFtype3", "T");
      } else
        editor.putString("mTFtype3", "F");
      editor.apply();
    }
    //return ret;
  }
}
```

Thinking/Feeling (T/F) Book.java

```
package com.example.apprize;
public class TFBook {

  public static String mChoices[][] = {
      {" logical, thinking, questioning ", " empathetic,
feeling, accommodating "},
      {" candid, straight forward, frank ", " tactful, kind,
encouraging "},
      {" firm, tend to criticize, hold the line ", " gentle, tend
to appreciate, conciliate "},
      {" tough-minded ", " tender-hearted, merciful "},
      {" matter of fact, issue-oriented ", " sensitive, people-
oriented, compassionate "}
  };

  public static String[] questions = new String [] {
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude."
  };

  public static String getChoice1(int a) {
    String choice0 = mChoices[a][0];
    return choice0;
  }


  public static String getChoice2(int a) {
    String choice1 = mChoices[a][1];
    return choice1;
  }

  public static int[] images = new int [] {
      R.drawable.feeling, R.drawable.kind,
R.drawable.gentle,
      R.drawable.hearts, R.drawable.compassion
  };
```

```
  public static String[] description = new String[] {

        "A 'Thinker' makes decisions in a rational, logical,
impartial manner, based on what they believe to be fair and
correct by pre-defined rules of behavior",
        "A 'Feeler' makes decisions on the individual case, in a
subjective manner based on what they believe to be right
within their own value systems"
  };
}
```

Judging/Perceiving (J/P) Test Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import com.google.firebase.database.FirebaseDatabase;
import java.util.ArrayList;

public class JPTestActivity extends AppCompatActivity {

  private TextView mQuestionNumberView, mQuestion;
  private ImageView mImageView;
  private Button mTrueButton, mFalseButton;
  int jcounter;
  int pcounter;
  private int m = 16;
  private int mQuestionNumber = 0;

   @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_jptest);
    final ArrayList<String> dlist = new ArrayList<>();
    FirebaseDatabase database =
FirebaseDatabase.getInstance();
    mQuestionNumberView = findViewById(R.id.points);
    mImageView = findViewById(R.id.imageView);
    mQuestion = findViewById(R.id.question);
    mTrueButton = findViewById(R.id.aButton);
    mFalseButton = findViewById(R.id.bButton);
    updateQuestion();
    updateType();

    //Logic for true button
    mTrueButton.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {

        if(mQuestionNumber == 0 || mQuestionNumber ==
1 || mQuestionNumber == 2 || mQuestionNumber == 3 ||
mQuestionNumber == 4) {
          updateType();
          m++;
          dlist.add("J");

          //perform check before you update the question
          if (mQuestionNumber ==
JPBook.questions.length) {
            Intent intent = new Intent(JPTestActivity.this,
MainTypeResultActivity.class);
            intent.putExtra("finalType4", dlist);
            startActivity(intent);
          } else {
            updateQuestion();
            updateType();
          }
          result();
        }
        else {

          //perform check before you update the question
          if (mQuestionNumber ==
JPBook.questions.length) {
            Intent intent = new Intent(JPTestActivity.this,
MainTypeResultActivity.class);
            intent.putExtra("finalType4", dlist);
            startActivity(intent);
          } else {
            updateQuestion();
            updateType();
          }
        }
      }
    });

    //Logic for false button
    mFalseButton.setOnClickListener(new
View.OnClickListener() {
      @Override
      public void onClick(View view) {

        if(mQuestionNumber == 0 || mQuestionNumber ==
1 || mQuestionNumber == 2 || mQuestionNumber == 3 ||
mQuestionNumber == 4) {
          updateType();
          m++;
          dlist.add("P");

          //perform check before you update the question
          if (mQuestionNumber ==
JPBook.questions.length) {
            Intent intent = new Intent(JPTestActivity.this,
MainTypeResultActivity.class);
            intent.putExtra("finalType4", dlist);
            startActivity(intent);
          } else {
            updateQuestion();
            updateType();
          }
          result();
        }
        else {
```

```
            //perform check before you update the question
            if (mQuestionNumber ==
JPBook.questions.length) {
              Intent intent = new Intent(JPTestActivity.this,
MainTypeResultActivity.class);
              intent.putExtra("finalType4", dlist);
              startActivity(intent);
            } else {
              updateQuestion();
              updateType();
            }
          }
        }
      });
  }

  private void updateType() {

    mQuestionNumberView.setText("" + m);
  }

  private void updateQuestion() {


mImageView.setImageResource(JPBook.images[mQuestion
Number]);

mTrueButton.setText(JPBook.getChoice1(mQuestionNumbe
r));

mFalseButton.setText(JPBook.getChoice2(mQuestionNumbe
r));

mQuestion.setText(JPBook.questions[mQuestionNumber]);
    mQuestionNumber++;
  }

  public void result (){
    if (m >= 5) {
      SharedPreferences pref;
      SharedPreferences.Editor editor;

      pref = getSharedPreferences("Apprize",
MODE_PRIVATE);
      editor = pref.edit();

      if (jcounter > pcounter) {
        editor.putString("mJPtype4", "J");
      } else
        editor.putString("mJPtype4", "P");
      editor.apply();
    }
    //return ret;
  }
}
```

Judging/Perceiving (J/P) Book.java

```
package com.example.apprize;
public class JPBook {

  public static String mChoices[][] = {
      {" organized, orderly ", "flexible, adaptable "},
      {" plan, schedule ", " unplanned, spontaneous "},
      {" regulated, structured ", " easygoing, “live” and “let
live” "},
      {" preparation, plan ahead ", " go with the flow, adapt
as you go "},
      {" control, govern ", " latitude, freedom "}


  };

  public static String[] questions = new String [] {
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude.",
      "Pick an answer that best describe your personality or
attitude."
  };

  public static String getChoice1(int a) {
    String choice0 = mChoices[a][0];
    return choice0;
  }

  public static String getChoice2(int a) {
    String choice1 = mChoices[a][1];
    return choice1;
  }

  public static int[] images = new int [] {
      R.drawable.flexible, R.drawable.plan, R.drawable.live,
R.drawable.prepare,
      R.drawable.control
  };

  public static String[] description = new String[] {
      "Judging people think sequentially. They value order
and organization. Their lives are scheduled and structured.
Judging people seek closure and enjoy completing tasks.",
      "Perceivers are adaptable and flexible. They are
random thinkers who prefer to keep their options open.
Perceivers thrive with the unexpected and are open to
change. They are spontaneous and often juggle several
projects at once."
  };
}
```

Main Type Result Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
```

```
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import java.util.ArrayList;

public class    MainTypeResultActivity extends
AppCompatActivity {

   TextView mFinalType, mTitle, mFinalDesc;
   Button mNextButton;
   Context c;
   SharedPreferences pref;
   SharedPreferences.Editor editor;
   public int s1,s2,s3,s4;
   String ftype, mEItype, mSNtype, mTFtype, mJPtype;

   @Override
   protected void onCreate(Bundle savedInstanceState) {

      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main_type_result);
      mFinalType = findViewById(R.id.final_ptypeTv);
      mTitle = findViewById(R.id.final_ptitleTv);
      mFinalDesc = findViewById(R.id.final_pdescTv);
      mNextButton = findViewById(R.id.next);

      /*Bundle bundle1 = getIntent().getExtras();
      ArrayList<String> type1 =
bundle1.getStringArrayList("finalType1");
      Bundle bundle2 = getIntent().getExtras();
      ArrayList<String> type2 =
bundle2.getStringArrayList("finalType2");
      Bundle bundle3 = getIntent().getExtras();
      ArrayList<String> type3 =
bundle3.getStringArrayList("finalType3");
      Bundle bundle4 = getIntent().getExtras();
      ArrayList<String> type4 =
bundle4.getStringArrayList("finalType4");*/


      /*pref =
getSharedPreferences("Apprize",MODE_PRIVATE);
      //ftype = pref.getString("mtype1 + mtype2 + mtype3 +
mtype4",ftype);
      mEItype = pref.getString("mEItype1", mEItype);
      mSNtype = pref.getString("mSNtype2", mSNtype);
      mTFtype = pref.getString("mTFtype3", mTFtype);
      mJPtype = pref.getString("mJPtype4", mJPtype);*/

      //mFinalType.setText("ISFP");
      //mTitle.setText(PTypesBook.title[15]);
      //mFinalDesc.setText(PTypesBook.description[15]);

      /*if (mFinalType.toString() == "ESTJ") {

         mTitle.setText(PTypesBook.title[0]);
         mFinalDesc.setText(PTypesBook.description[0]);

      }
      else if (mFinalType.toString() == "ESTP") {

         mTitle.setText(PTypesBook.title[1]);
         mFinalDesc.setText(PTypesBook.description[1]);

      }
      else if (mFinalType.toString() == "ESFJ") {

         mTitle.setText(PTypesBook.title[2]);
         mFinalDesc.setText(PTypesBook.description[2]);

      }
      else if (mFinalType.toString() == "ENFJ") {

         mTitle.setText(PTypesBook.title[3]);
         mFinalDesc.setText(PTypesBook.description[3]);

      }
      else if (mFinalType.toString() == "ESFP") {

         mTitle.setText(PTypesBook.title[4]);
         mFinalDesc.setText(PTypesBook.description[4]);

      }
      else if (mFinalType.toString() == "ENFP") {

         mTitle.setText(PTypesBook.title[5]);
         mFinalDesc.setText(PTypesBook.description[5]);

      }
      else if (mFinalType.toString() == "ENTJ") {

         mTitle.setText(PTypesBook.title[6]);
         mFinalDesc.setText(PTypesBook.description[6]);

      }
      else if (mFinalType.toString() == "ENTP") {

         mTitle.setText(PTypesBook.title[7]);
         mFinalDesc.setText(PTypesBook.description[7]);

      }
      else if (mFinalType.toString() == "ISTJ") {

         mTitle.setText(PTypesBook.title[8]);
         mFinalDesc.setText(PTypesBook.description[8]);

      }
      else if (mFinalType.toString() == "INFJ") {

         mTitle.setText(PTypesBook.title[9]);
         mFinalDesc.setText(PTypesBook.description[9]);

      }
      else if (mFinalType.toString() == "INTJ") {

         mTitle.setText(PTypesBook.title[10]);
         mFinalDesc.setText(PTypesBook.description[10]);

      }
      else if (mFinalType.toString() == "ISTP") {

         mTitle.setText(PTypesBook.title[11]);
```

```
        mFinalDesc.setText(PTypesBook.description[11]);

    }
    else if (mFinalType.toString() == "INFP") {

        mTitle.setText(PTypesBook.title[12]);
        mFinalDesc.setText(PTypesBook.description[12]);

    }
    else if (mFinalType.toString() == "INTP") {

        mTitle.setText(PTypesBook.title[13]);
        mFinalDesc.setText(PTypesBook.description[13]);

    }
    else if (mFinalType.toString() == "ISFJ") {

        mTitle.setText(PTypesBook.title[14]);
        mFinalDesc.setText(PTypesBook.description[14]);

    }
    else {

        mTitle.setText(PTypesBook.title[15]);
        mFinalDesc.setText(PTypesBook.description[15]);

    }*/


    mNextButton.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            startActivity(new
Intent(MainTypeResultActivity.this,
DashboardActivity.class));
            MainTypeResultActivity.this.finish();
            editor.remove("mEItype1"); // will delete key name
            editor.remove("mSNtype2");
            editor.remove("mTFtype3"); // will delete key
name
            editor.remove("mJPtype4");
            editor.commit();
        }
    });
  }
```

Dashboard Activity

```
package com.example.apprize;
import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentTransaction;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
import
com.google.android.material.bottomnavigation.BottomNavi
gationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class DashboardActivity extends AppCompatActivity {

  //firebase auth
  FirebaseAuth firebaseAuth;
  SharedPreferences pref;
  SharedPreferences.Editor editor;
  ActionBar actionBar;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_dashboard);

    //Action bar and its title
    actionBar = getSupportActionBar();
    actionBar.setTitle("Profile");

    //init
    firebaseAuth = FirebaseAuth.getInstance();

    //bottom navigation
    BottomNavigationView navigationView =
findViewById(R.id.navigation);

navigationView.setOnNavigationItemSelectedListener(select
edListener);
    actionBar.setTitle("Profile");//change actionbar title
    ProfileFragment fragment2 = new ProfileFragment();
    FragmentTransaction ft2 =
getSupportFragmentManager().beginTransaction();
    ft2.replace(R.id.content, fragment2,"");
    ft2.commit();
  }

  private
BottomNavigationView.OnNavigationItemSelectedListener
selectedListener = new
BottomNavigationView.OnNavigationItemSelectedListener()
{
    @Override
    public boolean onNavigationItemSelected(@NonNull
MenuItem menuItem) {
      //handle moodlifter_items clicks
      switch (menuItem.getItemId()){
        case R.id.nav_apprize:
          //apprize fragment transaction
          actionBar.setTitle("Apprize");//change actionbar
title
          ApprizeFragment fragment1 = new
ApprizeFragment();
          FragmentTransaction ft1 =
getSupportFragmentManager().beginTransaction();
          ft1.replace(R.id.content, fragment1, "");
          ft1.commit();
          return true;
```

```
            case R.id.nav_profile:
                //profile fragment transaction
                actionBar.setTitle("Profile");//change actionbar
title
                ProfileFragment fragment2 = new
ProfileFragment();
                FragmentTransaction ft2 =
getSupportFragmentManager().beginTransaction();
                ft2.replace(R.id.content, fragment2,"");
                ft2.commit();
                return true;
            case R.id.nav_friends :
                //users fragment transaction
                actionBar.setTitle("Users");//change actionbar
title
                FriendsFragment fragment3 = new
FriendsFragment();
                FragmentTransaction ft3 =
getSupportFragmentManager().beginTransaction();
                ft3.replace(R.id.content, fragment3, "");
                ft3.commit();
                return true;
        }
        return false;
    }
  };

  private  void checkUserStatus(){
    //get current user
    FirebaseUser user = firebaseAuth.getCurrentUser();

    if (user != null) {
      //user is signed in stay here
      //set email of logged in user
    }
    else {
      //user not signed in, go to main activity
      startActivity(new Intent(DashboardActivity.this,
MainActivity.class));
      finish();
      editor.clear();
      editor.apply();
    }
  }

  @Override
  public void onBackPressed() {
    super.onBackPressed();
    finish();
  }

  @Override
  public void onStart() {
    //check on start of app
    checkUserStatus();
    super.onStart();
  }

  //inflate options menu
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    //inflate menu
    getMenuInflater().inflate(R.menu.menu_main,menu);
    return super.onCreateOptionsMenu(menu);
  }

  //handle menu moodlifter_items clicks
  @Override
  public boolean onOptionsItemSelected(@NonNull
MenuItem item) {
    //get moodlifter_items id
    int id = item.getItemId();
    if (id == R.id.action_logout){
      firebaseAuth.signOut();
      checkUserStatus();
    }
    return super.onOptionsItemSelected(item);
  }
}
```

Profile Fragment

```
package com.example.apprize;
import android.Manifest;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.net.Uri;
import android.os.Bundle;
import androidx.annotation.ContentView;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.Fragment;
import android.provider.MediaStore;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import
com.google.android.material.floatingactionbutton.FloatingA
ctionButton;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
```

```
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;
import com.squareup.picasso.Picasso;
import org.w3c.dom.Text;
import java.security.Key;
import java.util.HashMap;
import static android.app.Activity.RESULT_OK;
import static
com.google.firebase.storage.FirebaseStorage.getInstance;

public class ProfileFragment extends Fragment {

   //firebase
   FirebaseAuth firebaseAuth;
   FirebaseUser user;
   FirebaseDatabase firebaseDatabase;
   DatabaseReference databaseReference;

   //storage
   StorageReference storageReference;

   //path where images of user profile and cover will be
stored
   String storagePath = "Users_Profile_Cover_Imgs/";

   //progress dialog
   ProgressDialog pd;

   //permissions constants
   private static final int CAMERA_REQUEST_CODE = 100;
   private static final int STORAGE_REQUEST_CODE = 200;
   private static final int IMAGE_PICK_GALLERY_CODE = 300;
   private static final int IMAGE_PICK_CAMERA_CODE = 400;

   //arrays of permission to be requested
   String cameraPermissions[];
   String storagePermissions[];

   //uri of picked image
   Uri image_uri;

   //for checking profile or cover photo
   String profileOrCoverPhoto;

   //views from xml
   ImageView avatarIv, coverIv;
   TextView nameTv, emailTv, quoteTv;
   FloatingActionButton fab;
   TextView descsc;

   public ProfileFragment() {
      // Required empty public constructor
   }

   @Override
   public View onCreateView(LayoutInflater inflater,
ViewGroup container,
                            Bundle savedInstanceState) {
      // Inflate the layout for this fragment
      View view = inflater.inflate(R.layout.fragment_profile,
container, false);

      //init firebase
      firebaseAuth = FirebaseAuth.getInstance();
      user = firebaseAuth.getCurrentUser();
      firebaseDatabase = FirebaseDatabase.getInstance();
      databaseReference =
firebaseDatabase.getReference("Users");
      storageReference =
getInstance().getReference();//firebase storage reference

      //init arrays of permissions
      cameraPermissions = new String[]
{Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE};
      storagePermissions = new String[]
{Manifest.permission.WRITE_EXTERNAL_STORAGE};

      //init views
      avatarIv = view.findViewById(R. id.avatarIv);
      coverIv = view.findViewById(R.id.coverIv);
      nameTv = view.findViewById(R. id.nameTv);
      emailTv = view.findViewById(R. id.emailTv);
      quoteTv = view.findViewById(R. id.quoteTv);
      fab = view.findViewById(R.id.fab);
      pd = new ProgressDialog(getActivity());
      descsc = view.findViewById(R.id.profiledescTv);
      descsc.setText("ISFP");

      descsc.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {

            Intent intent = new Intent(getActivity(),
PersonalityActivity.class);
            getActivity().startActivity(intent);

         }
      });

      Query query =
databaseReference.orderByChild("email").equalTo(user.getE
mail());
      query.addValueEventListener(new ValueEventListener()
{
         @Override
         public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {

            //check until required data get
            for (DataSnapshot ds: dataSnapshot.getChildren()){
               //get data
               String name = ""+ ds.child("name").getValue();
               String email = ""+ ds.child("email").getValue();
               String quote = ""+ ds.child("quote").getValue();
               String image= ""+ ds.child("image").getValue();
               String cover= ""+ ds.child("cover").getValue();

               //set data
```

```
                nameTv.setText(name);
                emailTv.setText(email);
                quoteTv.setText(quote);
                try {
                    //if image is received then set
                    Picasso.get().load(image).into(avatarIv);
                }
                catch (Exception e) {
                    //if there is any exceotion while getting image
then set default

Picasso.get().load(R.drawable.ic_default_img_white).into(av
atarIv);
                }

                try {
                    //if image is received then set
                    Picasso.get().load(cover).into(coverIv);
                }
                catch (Exception e) {
                    //if there is any exceotion while getting image
then set default
                }
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }
    });

    //fab button click
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            showEditProfileDialog();
        }
    });

    return view;
  }

  private boolean checkStoragePermission(){
    //check if storage permission is enabled or not
    //return true if enabled
    //return false if not enabled
    boolean result =
ContextCompat.checkSelfPermission(getActivity(),
Manifest.permission.WRITE_EXTERNAL_STORAGE)
            == (PackageManager.PERMISSION_GRANTED);
    return result;
  }

  private void requestStoragePersmission(){
    //request runtime storage permission
    requestPermissions(storagePermissions,
STORAGE_REQUEST_CODE);
  }

  private boolean checkCameraPermission(){
    //check if storage permission is enabled or not
    //return true if enabled
    //return false if not enabled
    boolean result =
ContextCompat.checkSelfPermission(getActivity(),
Manifest.permission.CAMERA)
            == (PackageManager.PERMISSION_GRANTED);

    boolean result1 =
ContextCompat.checkSelfPermission(getActivity(),
Manifest.permission.WRITE_EXTERNAL_STORAGE)
            == (PackageManager.PERMISSION_GRANTED);
    return result && result1;
  }

  private void requestCameraPersmission(){
    //request runtime storage permission
    requestPermissions(cameraPermissions,
CAMERA_REQUEST_CODE);

  }

  private void showEditProfileDialog() {
    /*Show dialog containing options
    1) Edit Profile Picture
    2) Edit Cover Photo
    3) Edit Name
    4) Edit Quote*/

    //options to show in dialog
    String options[] = {"Edit Profile Picture","Edit Cover
Photo", "Edit Name", "Edit Quote"};

    //alert dialog
    AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());

    //set title
    builder.setTitle("Choose Action");

    //set items to dialog
    builder.setItems(options, new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {

            //handle dialog moodlifter_items clicks
            if (which == 0){
                //Edit Profile Clicked
                pd.setMessage("Updating Profile Picture...");
                profileOrCoverPhoto = "image";
                showImagePicDialog();
            }
            else if (which == 1){
                //Edit Cover Clicked
                pd.setMessage("Updating Cover Photo...");
                profileOrCoverPhoto = "cover";
                showImagePicDialog();
            }
```

```
            else if (which == 2){
                //Edit Name Clicked
                pd.setMessage("Updating Name...");
                showNameUpdateDialog("name");
            }
            else if (which == 3){
                //Edit Quote Clicked
                pd.setMessage("Updating Quote...");
                showNameUpdateDialog("quote");
            }
        }
    });
    //create and show dialog
    builder.create().show();
  }

  private void showNameUpdateDialog(final String key) {

    //custom dialog
    final AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());
    builder.setTitle("Update "+ key);

    //set layout of dialog
    LinearLayout linearLayout = new
LinearLayout(getActivity());
    linearLayout.setOrientation(LinearLayout.VERTICAL);
    linearLayout.setPadding(10,10,10,10);
    //add edit test
    final EditText editText = new EditText(getActivity());
    editText.setHint("Enter "+key);
    linearLayout.addView(editText);
    builder.setView(linearLayout);

    //add button in dialog to update
    builder.setPositiveButton("Update", new
DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface dialogInterface, int
i) {

        //input text from edit text
        String value = editText.getText().toString().trim();

        //validate if user has entered something or not
        if (!TextUtils.isEmpty(value)){
          pd.show();
          HashMap<String, Object> result = new
HashMap<>();
          result.put(key, value);


databaseReference.child(user.getUid()).updateChildren(resu
lt).addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void aVoid) {

              //updated, dismiss progress
              pd.dismiss();
              Toast.makeText(getActivity(), "Updated...",
Toast.LENGTH_SHORT).show();
            }
          }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {

              //failed, dismiss progress, get and show error
message
              pd.dismiss();
              Toast.makeText(getActivity(),
""+e.getMessage(), Toast.LENGTH_SHORT).show();
            }
          });
        }
        else {
          Toast.makeText(getActivity(), "Please enter"+key,
Toast.LENGTH_SHORT).show();
        }
      }
    });
    //add button in dialog to cancel
    builder.setNegativeButton("Cancel", new
DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface dialogInterface, int
i) {
        dialogInterface.dismiss();
      }
    });

    //create and show dialog
    builder.create().show();
  }

  private void showImagePicDialog() {
    //show dialog containing options Camera and Gallery to
pick the image
    //options to show in dialog
    String options[] = {"Camera", "Gallery"};

    //alert dialog
    AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity());

    //set title
    builder.setTitle("Pick Image from...");

    //set items to dialog
    builder.setItems(options, new
DialogInterface.OnClickListener() {
      @Override
      public void onClick(DialogInterface dialogInterface, int
i) {
        //handle dialog moodlifter_items clicks
        if (i == 0){
          //Camera Clicked
          if (!checkCameraPermission()){
            requestCameraPersmission();
          }
          else {
            pickFromCamera();
          }
```

```
            }
            else if (i == 1){
                //Gallery Clicked
                if (!checkStoragePermission()){
                    requestStoragePersmission();
                }
                else {
                    pickFromGallery();
                }
            }
        }
    });
    //create and show dialog
    builder.create().show();
}

@Override
public void onRequestPermissionsResult(int requestCode,
@NonNull String[] permissions,
                                       @NonNull int[] grantResults) {

    //This method called when user press allow or deny
from permission request dialog
    //here we will handle permission cases (allowed &
denied)

    switch (requestCode){
        case CAMERA_REQUEST_CODE:{

            //picking from camera, first check if camera and
storage permissions allowed or not

            if (grantResults.length > 0){
                boolean cameraAccepted = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                boolean writeStorageAccepted = grantResults[1]
== PackageManager.PERMISSION_GRANTED;
                if (cameraAccepted && writeStorageAccepted){
                    //permissions enabled
                    pickFromCamera();
                }
                else {
                    //permission denied
                    Toast.makeText(getActivity(), "Please enable
camera and storage permission",
Toast.LENGTH_SHORT).show();
                }
            }
        }
        break;
        case STORAGE_REQUEST_CODE:{

            //picking from camera, first check if storage
permissions allwed or not
            if (grantResults.length > 0){

                boolean writeStorageAccepted = grantResults[1]
== PackageManager.PERMISSION_GRANTED;
                if (writeStorageAccepted){
                    //permissions enabled
                    pickFromGallery();
                }
                else {
                    //permission denied
                    Toast.makeText(getActivity(), "Please enable
storage permission", Toast.LENGTH_SHORT).show();
                }
            }
        }
        break;
    }
}

@Override
public void onActivityResult(int requestCode, int
resultCode, @Nullable Intent data) {
    //this method will be called after picking image from
camera or gallery
    if (resultCode == RESULT_OK){

        if (requestCode == IMAGE_PICK_GALLERY_CODE){
            //image is picked from gallery, get uri of image
            image_uri = data.getData();
            uploadProfileCoverPhoto(image_uri);

        }
        if (requestCode == IMAGE_PICK_CAMERA_CODE){

            //image is picked from camera, get uri of image
            uploadProfileCoverPhoto(image_uri);
        }
    }
    super.onActivityResult(requestCode, resultCode, data);
}

private void uploadProfileCoverPhoto(final Uri uri) {

    //show progress
    pd.show();

    //path and name of image to be stored in firebase
storage
    String filePathAndName = storagePath+ ""+
profileOrCoverPhoto+ ""+ user.getUid();

    StorageReference storageReference2nd =
storageReference.child(filePathAndName);

storageReference2nd.putFile(uri).addOnSuccessListener(ne
w OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot
taskSnapshot) {

            //image is uploaded to storage, now get its url and
store in user's database
            Task<Uri> uriTask =
taskSnapshot.getStorage().getDownloadUrl();
            while (!uriTask.isSuccessful());
            Uri downloadUri = uriTask.getResult();
```

```
            //check if image is uploaded or not and uri is
received
            if ( uriTask.isSuccessful()){

                //image uploaded
                //add/update url in user's database
                HashMap<String, Object> results = new
HashMap<>();
                results.put(profileOrCoverPhoto,
downloadUri.toString());


databaseReference.child(user.getUid()).updateChildren(resu
lts).addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void aVoid) {

                        //url in database of user is added succesfully
                        //dismiss progress bar
                        pd.dismiss();
                        Toast.makeText(getActivity(), "Image
Updated...", Toast.LENGTH_SHORT).show();
                    }
                }).addOnFailureListener(new OnFailureListener() {
                    @Override
                    public void onFailure(@NonNull Exception e) {

                        //error adding url in database of user
                        //dismiss progress bar
                        pd.dismiss();
                        Toast.makeText(getActivity(), "Error
Updating Image...", Toast.LENGTH_SHORT).show();
                    }
                });
            }
            else {
                //error
                pd.dismiss();
                Toast.makeText(getActivity(), "Some error
occured", Toast.LENGTH_SHORT).show();
            }
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {

            //there will some error(s), get and show error
message, dismiss progress dialog
            pd.dismiss();
            Toast.makeText(getActivity(), e.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    });
  }

  private void pickFromCamera() {

    //Intent of picking image from device camera
    ContentValues values = new ContentValues();
    values.put(MediaStore.Images.Media.TITLE, "Temp
Pic");
    values.put(MediaStore.Images.Media.DESCRIPTION,
"Temp Description");

    //put image uri
    image_uri =
getActivity().getContentResolver().insert(MediaStore.Images
.Media.EXTERNAL_CONTENT_URI, values);

    //intent to start camera
    Intent cameraIntent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    cameraIntent.putExtra(MediaStore.EXTRA_OUTPUT,
image_uri);
    startActivityForResult(cameraIntent,
IMAGE_PICK_CAMERA_CODE);
    }

  private void pickFromGallery() {
    //pick from gallery
    Intent galleryIntent = new Intent(Intent.ACTION_PICK);
    galleryIntent.setType("image/*");
    startActivityForResult(galleryIntent,
IMAGE_PICK_GALLERY_CODE);
  }
}
```

Apprize Fragment

```
package com.example.apprize;
import android.content.Intent;
import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;

public class ApprizeFragment extends Fragment {

  //buttons
  Button moodBtn, liftBtn, sentiBtn;

  public ApprizeFragment() {
    // Required empty public constructor
  }


  @Override
  public View onCreateView(LayoutInflater inflater,
ViewGroup container,
                           Bundle savedInstanceState) {
    // Inflate the layout for this fragment

    //init
    View rootView =
inflater.inflate(R.layout.fragment_apprize, container, false);
    moodBtn = (Button)
rootView.findViewById(R.id.moodlifterBtn);
    liftBtn = (Button) rootView.findViewById(R.id.upliftBtn);
    sentiBtn = (Button)
rootView.findViewById(R.id.sentimentsBtn);
```

```
        moodBtn.setOnClickListener(new
View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent intent = new Intent(getActivity(),
MoodActivities.class);
                getActivity().startActivity(intent);
            }
        });

        liftBtn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                Intent intent = new Intent(getActivity(),
PersonalityActivity.class);
                getActivity().startActivity(intent);
            }
        });

        sentiBtn.setOnClickListener(new View.OnClickListener()
{

            @Override
            public void onClick(View arg0) {
                Intent intent = new
Intent(getActivity(),SentimentsActivity.class);
                getActivity().startActivity(intent);
            }
        });
        return rootView;
        //return inflater.inflate(R.layout.fragment_apprize,
container, false);
    }
}
```

Personality Types Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;
import android.os.Bundle;
import android.widget.LinearLayout;

public class PersonalityActivity extends AppCompatActivity {

    private ViewPager mSlideViewPager;
    private LinearLayout mDotLayout;
    private SliderAdapter sliderAdapter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_personality);
        mSlideViewPager = (ViewPager)
findViewById(R.id.slideViewPager);
        mDotLayout = (LinearLayout)
findViewById(R.id.dotsLayout);
        sliderAdapter = new SliderAdapter(this);
        mSlideViewPager.setAdapter(sliderAdapter);
    }
}
```

Slider Adapter.java

```
package com.example.apprize;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.viewpager.widget.PagerAdapter;
import java.net.ConnectException;

public class SliderAdapter extends PagerAdapter {

    Context context;
    LayoutInflater layoutInflater;

    public SliderAdapter(Context context){

        this.context = context;
    }

    //Arrays
    public int[] slide_images = {
            R.drawable.istjwoodpecker,
            R.drawable.enfjeagle,
            R.drawable.enfjpeacock,
            R.drawable.enfpbluebird,
            R.drawable.entpcockatoo,
            R.drawable.esfjswan,
            R.drawable.esfpbudgie,
            R.drawable.estjpenguin,
            R.drawable.estpkingfisher,
            R.drawable.infjhummingbird,
            R.drawable.infpdove,
            R.drawable.intjowl,
            R.drawable.intpraven,
            R.drawable.isfjweaverbird,
            R.drawable.isfprobin,
            R.drawable.istpcuckoo
    };

    public String[] slide_personality = {

            "ISFP",
            "ENTJ",
            "ENFJ",
            "ENFP",
            "ENTP",
            "ESFJ",
            "ESTJ",
            "ESFP",
            "ESTP",
            "INTJ",
```

```
        "INFJ",
        "INFP",
        "INTP",
        "ISFJ",
        "ISTJ",
        "ISTP"
    };

    public String[] slide_label = {
        "The Inspector",
        "The Commander",
        "The Giver",
        "The Champion",
        "The Visionary",
        "The Provider",
        "The Performer",
        "The Supervisor",
        "The Doer",
        "The Counselor",
        "The Idealist",
        "The Mastermind",
        "The Thinker",
        "The Nurturer",
        "The Composer",
        "The Craftsman",
    };

    public String[] slide_birds = {

        "(WOODPECKER)",
        "(EAGLE)",
        "(PEACOCK)",
        "(BLUEBIRD)",
        "(COCKATOO)",
        "(SWAN)",
        "(BUDGIE)",
        "(PENGUIN)",
        "(KINGFISHER)",
        "(HUMMING BIRD)",
        "(DOVE)",
        "(OWL)",
        "(RAVEN)",
        "(WEAVERBIRD)",
        "(ROBIN)",
        "(CUCKOO)"
    };

    public String[] slide_desc = {
        "ISFPs are introverts that do not seem like introverts. It is because even if they have difficulties connecting to other people at first, they become warm, approachable, and friendly eventually. They are fun to be with and very spontaneous, which makes them the perfect friend to tag along in whatever activity, regardless if planned or unplanned. ISFPs want to live their life to the fullest and embrace the present, so they make sure they are always out to explore new things and discover new experiences.",
        "An ENTJ's primary mode of living focuses on external aspects and all things are dealt with rationally and logically. Their secondary mode of operation is internal, where intuition and reasoning take effect. ENTJs are natural born leaders among the 16 personality types and like being in charge. They seem to have a natural gift for leadership, making decisions, and considering options and ideas quickly yet carefully. They are "take charge" people who do not like to sit still.",
        "ENFJs are people-focused individuals. They are extroverted, idealistic, charismatic, outspoken, highly principled and ethical, and usually know how to connect with others no matter their background or personality. Mainly relying on intuition and feelings, they tend to live in their imagination rather than in the real world. Instead of focusing on living in the "now" and what is currently happening, ENFJs tend to concentrate on the abstract and what could possibly happen in the future",
        "ENFPs have an Extraverted, Intuitive, Feeling and Perceiving personality. This personality type is highly individualistic and Champions strive toward creating their own methods, looks, actions, habits, and ideas — they do not like cookie cutter people and hate when they are forced to live inside a box. They like to be around other people and have a strong intuitive nature when it comes to themselves and others. They operate from their feelings most of the time, and they are highly perceptive and thoughtful.",
        "Those with the ENTP personality are some of the rarest in the world, which is completely understandable. Although they are extroverts, they don't enjoy small talk and may not thrive in many social situations, especially those that involve people who are too different from the ENTP. ENTPs are intelligent and knowledgeable need to be constantly mentally stimulated. They have the ability to discuss theories and facts in extensive detail. They are logical, rational, and objective in their approach to information and arguments.",
        "ESFJs are the stereotypical extroverts. They are social butterflies, and their need to interact with others and make people happy usually ends up making them popular. The ESFJ usually tends to be the cheerleader or sports hero in high school and college. Later on in life, they continue to revel in the spotlight, and are primarily focused on organizing social events for their families, friends and communities. ESFJ is a common personality type and one that is liked by many people.",
        "ESFPs have an Extraverted, Observant, Feeling and Perceiving personality, and are commonly seen as Entertainers. Born to be in front of others and to capture the stage, ESFPs love the spotlight. ESFPs are thoughtful explorers who love learning and sharing what they learn with others. ESFPs are "people people" with strong interpersonal skills. They are lively and fun, and enjoy being the center of attention. They are warm, generous, and friendly, sympathetic and concerned for other people's well-being.",
        "ESTJs are organized, honest, dedicated, dignified, traditional, and are great believers of doing what they believe is right and socially acceptable. Though the paths towards "good" and "right" are difficult, they are glad to take their place as the leaders of the pack. They are the epitome of good citizenry. People look to ESTJs for guidance and counsel, and ESTJs are always happy that they are approached for help.",
```

```
        "ESTPs have an Extraverted, Sensing, Thinking, and Perceptive personality. ESTPs are governed by the need for social interaction, feelings and emotions, logical processes and reasoning, along with a need for freedom. Theory and abstracts don't keep ESTP's interested for long. ESTPs leap before they look, fixing their mistakes as they go, rather than sitting idle or preparing contingency plans.",
        "INFJs are visionaries and idealists who ooze creative imagination and brilliant ideas. They have a different, and usually more profound, way of looking at the world. They have a substance and depth in the way they think, never taking anything at surface level or accepting things the way they are. Others may sometimes perceive them as weird or amusing because of their different outlook on life.",
        "INFPs, like most introverts, are quiet and reserved. They prefer not to talk about themselves, especially in the first encounter with a new person. They like spending time alone in quiet places where they can make sense of what is happening around them. They love analyzing signs and symbols, and consider them to be metaphors that have deeper meanings related to life. They are lost in their imagination and daydreams, always drowned in the depth of their thoughts, fantasies, and ideas.",
        "INTJs, as introverts, are quiet, reserved, and comfortable being alone. They are usually self-sufficient and would rather work alone than in a group. Socializing drains an introvert's energy, causing them to need to recharge. INTJs are interested in ideas and theories. When observing the world they are always questioning why things happen the way they do. They excel at developing plans and strategies, and don't like uncertainty.",
        "INTPs are well known for their brilliant theories and unrelenting logic, which makes sense since they are arguably the most logical minded of all the personality types. People of this personality type aren't interested in practical, day-to-day activities and maintenance, but when they find an environment where their creative genius and potential can be expressed, there is no limit to the time and energy INTPs will expend in developing an insightful and unbiased solution.",
        "ISFJs are philanthropists and they are always ready to give back and return generosity with even more generosity. The people and things they believe in will be upheld and supported with enthusiasm and unselfishness. ISFJs are warm and kind-hearted. They value harmony and cooperation, and are likely to be very sensitive to other people's feelings. People value the ISFJ for their consideration and awareness, and their ability to bring out the best in others.",
        "At first glance, ISTJs are intimidating. They appear serious, formal, and proper. They also love traditions and old-school values that uphold patience, hard work, honor, and social and cultural responsibility. They are reserved, calm, quiet, and upright. These traits result from the combination of I, S, T, and J, a personality type that is often misunderstood.",
        "ISTPs are mysterious people who are usually very rational and logical, but also quite spontaneous and enthusiastic. Their personality traits are less easily recognizable than those of other types, and even people who know them well can't always anticipate their reactions. Deep down, ISTPs are spontaneous, unpredictable individuals, but they hide those traits from the outside world, often very successfully."
    };

    @Override
    public int getCount() {
        return slide_personality.length;
    }

    @Override
    public boolean isViewFromObject(@NonNull View view, @NonNull Object object) {
        return view == (RelativeLayout) object;
    }

    @NonNull
    @Override
    public Object instantiateItem(@NonNull ViewGroup container, int position) {

        layoutInflater = (LayoutInflater) context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
        View view = layoutInflater.inflate(R.layout.slide_layout, container, false);
        TextView slidePersonality = (TextView) view.findViewById(R.id.ptTextView);
        ImageView slideImageView = (ImageView) view.findViewById(R.id.birdIV);
        TextView slideLabel = (TextView) view.findViewById(R.id.labelTV);
        TextView slideBird = (TextView) view.findViewById(R.id.birdTV);
        TextView slideDesc = (TextView) view.findViewById(R.id.descTV);
        slidePersonality.setText(slide_personality[position]);

        slideImageView.setImageResource(slide_images[position]);
        slideLabel.setText(slide_label[position]);
        slideBird.setText(slide_birds[position]);
        slideDesc.setText(slide_desc[position]);
        container.addView(view);

        return view;
    }

    @Override
    public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {

        container.removeView((RelativeLayout)object);
    }
}
```

Cards.java

```
package com.example.apprize;

public class cards {
```

```
  private String userID;
  private String name;

  public cards(String userID, String name) {
    this.userID = userID;
    this.name = name;
  }

  public String getUserID() {
    return userID;
  }

  public void setUserID(String userID) {
    this.userID = userID;
  }

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }
}
```

MoodLifter Activity

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class MoodLifterActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_mood_lifter);
  }
}
```

Mood Activities

```
package com.example.apprize;
import androidx.appcompat.app.AppCompatActivity;
import androidx.viewpager.widget.ViewPager;
import android.animation.ArgbEvaluator;
import android.os.Bundle;
import android.view.Display;
import java.util.ArrayList;
import java.util.List;

public class MoodActivities extends AppCompatActivity {

  ViewPager viewPager;
  Adapter adapter;
  Integer[] colors = null;
  List<Model> models;
  ArgbEvaluator argbEvaluator = new ArgbEvaluator();

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_mood_activities);
    models = new ArrayList<>();
    models.add(new Model(R.drawable.smiletwo, "1. Smile", "The act of smiling really can turn a frown upside down."));
    models.add(new Model(R.drawable.jump, "2. Jump Around", "Get happy-making endorphins pumping fast with some jumping jacks, jump rope, or random flailing around."));
    models.add(new Model(R.drawable.scent, "3. Sniff certain scents", "Inhaling the scent of orange (or essential orange oil) or lavender can reduce anxiety and improve mood."));
    models.add(new Model(R.drawable.gum, "4. Chew gum", "The repetitive action of gnawing on gum can promote relaxation and reduce anxiety and stress."));
    models.add(new Model(R.drawable.flowers, "5. Ogle some flowers", "Studies find flowers provide an instant—and lasting—mood boost. Bonus: they can also make us more productive."));
    models.add(new Model(R.drawable.chocolate, "6. Eat some chocolate", "As if we needed a reason other than delicious: Eating chocolate can make us feel happy."));
    models.add(new Model(R.drawable.best, "7. See your best self", "Let's be honest: None of us are exactly the person we want to be all the time. But imagining our "ideal" selves—calm, confident, movin' like Jagger—can make us feel better."));
    models.add(new Model(R.drawable.blessings, "8. Count your blessings", "Think about or write down what you're thankful for. Expressing gratitude creates an instant mood boost."));
    models.add(new Model(R.drawable.snuggle, "9. Snuggle up", "Researchers found there's something about contact with soft things that just makes us feel better."));
    models.add(new Model(R.drawable.nice, "10. Do something nice for someone", "Yep, being nice can help us feel nicer. "));
    models.add(new Model(R.drawable.song, "11. Listen to a happy song", "It's quick; it's easy; it's an instant mood-lifter. Sing along (perfect pitch not required) for extra benefit."));
    models.add(new Model(R.drawable.quiter, "12. Go somewhere quite", "Taking a few minutes to sit in a quiet space with no external stimulation can do wonders for a bad mood. "));
    models.add(new Model(R.drawable.cuddle, "13. Cuddle", "Physical touch can decrease stress, make us feel happier, and even improve our health. Even a quick hug with a friend or acquaintance can yield benefits."));
    models.add(new Model(R.drawable.goal, "14. Achieve a goal", "Even small successes can have big mood payoffs."));
    models.add(new Model(R.drawable.massage, "15. Have a massage", "A quick rubdown (focus on the neck, shoulders, lower back, and feet) can improve mood and release stress."));
    models.add(new Model(R.drawable.meditate, "16. Meditate", "Meditation is a quick, effective way to chill out
```

```
and improve our outlook, and it might even make us
smarter. "));
        models.add(new Model(R.drawable.laughter, "17.
Laugh", "Laughter can cheer us up and decrease anxiety—
and the best news is it doesn't have to be "genuine" to have
a positive effect."));
        models.add(new Model(R.drawable.newnew, "18. Do
something new", "Adding something small to a normal
routine can brighten up a day."));
        models.add(new Model(R.drawable.dress, "19. Dress
up", "Speaking of clothing: Buying new garb can amp up
mood, but a person doesn't have to drop cash to reap
clothes' benefits. "));
        models.add(new Model(R.drawable.miracle, "20. Notice
small miracles", "Cultivating positivity and a sense of wonder
can build positive outlook."));
        models.add(new Model(R.drawable.call, "21. Call an
upbeat friend", "If you want to be happy and calm, spend
time around calm, happy people. If you only have a few
minutes, call one of them."));
        models.add(new Model(R.drawable.declutter, "22. De-
clutter", "Getting organized can help us feel instantly calmer.
Just five to ten minutes is enough to tackle a small
project."));
        models.add(new Model(R.drawable.distrations, "23.
Invite Distractions", "Step away from worries for a few
minutes and get absorbed in something neutral. "));
        models.add(new Model(R.drawable.eat, "24. Eat a
positive food", "We are what we eat, so step away from the
unhappy meal. Instead, try out these meals to boost your
mood. "));
        models.add(new Model(R.drawable.vent, "25. Vent to a
friend", "So long as it doesn't go on and on (and on), venting
can actually make us feel better about our problems."));
        models.add(new Model(R.drawable.celeberate, "26.
Celebrate good times", "Look at happy photos or spend a
minute or so thinking back on positive memories - nostalgia
can trigger happiness."));
        models.add(new Model(R.drawable.sun, "27. Get some
sun", "A boost of vitamin D can keep the blues at bay. Head
outside for a brisk walk around the block. "));
        models.add(new Model(R.drawable.yoga, "28. Do some
yoga", "A few hip openers might be the answer to a brighter
day. Don't forget to breathe deep."));
        models.add(new Model(R.drawable.rearrang, "29.
Rearrange some stuff", "Changing an environment can help
us feel refreshed, enabling us to bust out of a negative
mood."));
        models.add(new Model(R.drawable.heart, "30. Love
yourself", "The best way of uplifting your mood is by
accepting yourself and the things you can't change. Always
remember to love yourself everyday."));

        adapter = new Adapter(models, this);
        viewPager = findViewById(R.id.viewPager);
        viewPager.setAdapter(adapter);
        viewPager.setPadding(130, 0,130,0);

        Integer[] colors_temp = {
                getResources().getColor(R.color.color1),
                getResources().getColor(R.color.color2),
                getResources().getColor(R.color.color3),
                getResources().getColor(R.color.color4),
                getResources().getColor(R.color.color5),
                getResources().getColor(R.color.color6),
                getResources().getColor(R.color.color7),
                getResources().getColor(R.color.color8),
                getResources().getColor(R.color.color9),
                getResources().getColor(R.color.color10),
                getResources().getColor(R.color.color11),
                getResources().getColor(R.color.color12),
                getResources().getColor(R.color.color13),
                getResources().getColor(R.color.color14),
                getResources().getColor(R.color.color15),
                getResources().getColor(R.color.color16),
                getResources().getColor(R.color.color17),
                getResources().getColor(R.color.color18),
                getResources().getColor(R.color.color19),
                getResources().getColor(R.color.color20),
                getResources().getColor(R.color.color21),
                getResources().getColor(R.color.color22),
                getResources().getColor(R.color.color23),
                getResources().getColor(R.color.color24),
                getResources().getColor(R.color.color25),
                getResources().getColor(R.color.color26),
                getResources().getColor(R.color.color27),
                getResources().getColor(R.color.color28),
                getResources().getColor(R.color.color29),
                getResources().getColor(R.color.color30)
        };

        colors = colors_temp;

        viewPager.setOnPageChangeListener(new
ViewPager.OnPageChangeListener() {
            @Override
            public void onPageScrolled(int position, float
positionOffset,
                                       int positionOffsetPixels) {
                if (position < (adapter.getCount() - 1) && position <
(colors.length - 1)){
                    viewPager.setBackgroundColor(
                            (Integer)argbEvaluator.evaluate(
                                    positionOffset,
                                    colors[position],
                                    colors[position + 1]
                            )
                    );
                }
                else  {

viewPager.setBackgroundColor(colors[colors.length-1]);
                }
            }

            @Override
            public void onPageSelected(int position) {
            }

            @Override
            public void onPageScrollStateChanged(int state) {
```

```
        }
      });
  }
}
```

Model.java

```
package com.example.apprize;
public class Model {

  private int moodImageIv;
  private String moodTitleTv;
  private String moodDescriptionTv;

  public Model(int moodImageIv, String moodTitleTv, String
moodDescriptionTv) {

    this.moodImageIv = moodImageIv;
    this.moodTitleTv = moodTitleTv;
    this.moodDescriptionTv = moodDescriptionTv;
  }

  public int getMoodImageIv() {
    return moodImageIv;
  }

  public void setMoodImageIv(int moodImageIv) {
    this.moodImageIv = moodImageIv;
  }

  public String getMoodTitleTv() {
    return moodTitleTv;
  }

  public void setMoodTitleTv(String moodTitleTv) {
    this.moodTitleTv = moodTitleTv;
  }

  public String getMoodDescriptionTv() {
    return moodDescriptionTv;
  }

  public void setMoodDescriptionTv(String
moodDescriptionTv) {
    this.moodDescriptionTv = moodDescriptionTv;
  }
}
```

Sentiments Activity

```
package com.example.apprize;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.content.Intent;
import android.content.res.Resources;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.text.TextUtils;
import android.util.TypedValue;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ServerValue;
import com.google.firebase.database.ValueEventListener;
import java.util.HashMap;
import java.util.Map;

public class SentimentsActivity extends AppCompatActivity {

  private RecyclerView mNotesList;
  private GridLayoutManager gridLayoutManager;
  private DatabaseReference fNotesDatabase;
  private FirebaseAuth fAuth;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_sentiments);
    mNotesList = findViewById(R.id.main_note_list);
    gridLayoutManager = new GridLayoutManager(this, 1 ,
GridLayoutManager.VERTICAL, false);
    mNotesList.setHasFixedSize(true);
    mNotesList.setLayoutManager(gridLayoutManager);
    //gridLayoutManager.setReverseLayout(true);
    //gridLayoutManager.setStackFromEnd(true);
    mNotesList.addItemDecoration(new
GridSpacingItemDecoration(1, dpToPx(10), true));
    fAuth = FirebaseAuth.getInstance();

    if (fAuth.getCurrentUser() != null) {
      fNotesDatabase =
FirebaseDatabase.getInstance().getReference().child("Notes
").child(fAuth.getCurrentUser().getUid());
    }
    loadData();
  }

  @Override
  public void onStart() {
    super.onStart();
  }

  private void loadData() {
    Query query = fNotesDatabase.orderByValue();
```

```
    FirebaseRecyclerAdapter<NoteModel,
NoteViewHolder> firebaseRecyclerAdapter = new
FirebaseRecyclerAdapter<NoteModel, NoteViewHolder>(

        NoteModel.class,
        R.layout.single_note_layout,
        NoteViewHolder.class,
        query
    ) {
      @Override
      protected void populateViewHolder(final
NoteViewHolder viewHolder, NoteModel model, int
position) {
        final String noteId = getRef(position).getKey();


fNotesDatabase.child(noteId).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot
dataSnapshot) {

              if (dataSnapshot.hasChild("title") &&
dataSnapshot.hasChild("timestamp")) {
                String title =
dataSnapshot.child("title").getValue().toString();
                String timestamp =
dataSnapshot.child("timestamp").getValue().toString();
                viewHolder.setNoteTitle(title);
                //viewHolder.setNoteTime(timestamp);
                GetTimeAgo getTimeAgo = new
GetTimeAgo();

viewHolder.setNoteTime(getTimeAgo.getTimeAgo(Long.pars
eLong(timestamp), getApplicationContext()));


viewHolder.noteCard.setOnClickListener(new
View.OnClickListener() {
                  @Override
                  public void onClick(View view) {
                    Intent intent = new
Intent(SentimentsActivity.this, NewNoteActivity.class);
                    intent.putExtra("noteId", noteId);
                    startActivity(intent);
                  }
                });
              }
            }

            @Override
            public void onCancelled(DatabaseError
databaseError) {

            }
          });
        }
    };
    mNotesList.setAdapter(firebaseRecyclerAdapter);
  }
```

```
  @Override
  public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.menunote_main,
menu);
    return true;
  }

  @Override
  public boolean onOptionsItemSelected( MenuItem item) {

    super.onOptionsItemSelected(item);
    switch (item.getItemId()) {
      case R.id.main_new_note_btn:
        Intent newIntent = new
Intent(SentimentsActivity.this, NewNoteActivity.class);
        startActivity(newIntent);
        break;
    }

    return true;
  }

  // Converting dp to pixel

  private int dpToPx(int dp) {
    Resources r = getResources();
    return
Math.round(TypedValue.applyDimension(TypedValue.COMP
LEX_UNIT_DIP, dp, r.getDisplayMetrics()));
  }
}
```

New Note Activity

```
package com.example.apprize;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import android.content.Intent;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.text.TextUtils;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ServerValue;
import com.google.firebase.database.ValueEventListener;
import java.util.HashMap;
```

```
import java.util.Map;

public class NewNoteActivity extends AppCompatActivity {

   private Button btnCreate;
   private EditText etTitle, etContent;
   private FirebaseAuth fAuth;
   private DatabaseReference fNotesDatabase;
   private Menu mainMenu;
   private String noteID = "no";
   private boolean isExist;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_new_note);

      try {
         noteID = getIntent().getStringExtra("noteId");
         //Toast.makeText(this, noteID,
Toast.LENGTH_SHORT).show();
         if (!noteID.trim().equals("")) {
            isExist = true;
         } else {
            isExist = false;
         }
      } catch (Exception e) {
         e.printStackTrace();
      }
      btnCreate = findViewById(R.id.new_note_btn);
      etTitle = findViewById(R.id.new_note_title);
      etContent = findViewById(R.id.new_note_content);
      //mToolbar = findViewById(R.id.new_note_toolbar);
      //setSupportActionBar(mToolbar);

getSupportActionBar().setDisplayShowHomeEnabled(true);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
      fAuth = FirebaseAuth.getInstance();
      fNotesDatabase =
FirebaseDatabase.getInstance().getReference().child("Notes
").child(fAuth.getCurrentUser().getUid());
      btnCreate.setOnClickListener(new
View.OnClickListener() {
         @Override
         public void onClick(View view) {
            String title = etTitle.getText().toString().trim();
            String content =
etContent.getText().toString().trim();

            if (!TextUtils.isEmpty(title) &&
!TextUtils.isEmpty(content)){
               createNote(title,content);
            }else{
               Snackbar.make(view,"Fill empty
fields",Snackbar.LENGTH_SHORT).show();
            }
         }
      });
      putData();
   }

   private void putData() {

      if (isExist) {

fNotesDatabase.child(noteID).addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot
dataSnapshot) {
               if (dataSnapshot.hasChild("title") &&
dataSnapshot.hasChild("content")) {
                  String title =
dataSnapshot.child("title").getValue().toString();
                  String content =
dataSnapshot.child("content").getValue().toString();
                  etTitle.setText(title);
                  etContent.setText(content);
               }
            }

            @Override
            public void onCancelled(DatabaseError
databaseError) {

            }
         });
      }
   }
   private void createNote(String title, String content) {

      if (fAuth.getCurrentUser() != null) {
         if (isExist) {
            // UPDATE A NOTE
            Map updateMap = new HashMap();
            updateMap.put("title",
etTitle.getText().toString().trim());
            updateMap.put("content",
etContent.getText().toString().trim());
            updateMap.put("timestamp",
ServerValue.TIMESTAMP);

fNotesDatabase.child(noteID).updateChildren(updateMap);
            Toast.makeText(this, "Note updated",
Toast.LENGTH_SHORT).show();
            startActivity(new Intent(NewNoteActivity.this,
SentimentsActivity.class));
            finish();
         } else {

            // CREATE A NEW NOTE
            final DatabaseReference newNoteRef =
fNotesDatabase.push();
            final Map noteMap = new HashMap();
            noteMap.put("title", title);
            noteMap.put("content", content);
            noteMap.put("timestamp",
ServerValue.TIMESTAMP);

            Thread mainThread = new Thread(new Runnable() {
               @Override
```

```
            public void run() {

newNoteRef.setValue(noteMap).addOnCompleteListener(ne
w OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull
Task<Void> task) {
                      if (task.isSuccessful()) {
                        Toast.makeText(NewNoteActivity.this,
"Note added", Toast.LENGTH_SHORT).show();
                        startActivity(new
Intent(NewNoteActivity.this, SentimentsActivity.class));
                        finish();
                      } else {
                        Toast.makeText(NewNoteActivity.this,
"ERROR: " + task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
                      }
                    }
                  });
              }
            });
            mainThread.start();
          }
      } else {
          Toast.makeText(this, "USERS IS NOT SIGNED IN",
Toast.LENGTH_SHORT).show();
      }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

      super.onCreateOptionsMenu(menu);

getMenuInflater().inflate(R.menu.new_note_menu,menu);
      return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull
MenuItem item) {
      super.onOptionsItemSelected(item);

      switch (item.getItemId()) {
        case android.R.id.home:
          finish();
          break;
        case R.id.new_note_delete_btn:
          if (isExist) {
            deleteNote();
          } else {
            Toast.makeText(this, "Nothing to delete",
Toast.LENGTH_SHORT).show();
          }
          break;
      }
      return true;
    }

    private void deleteNote() {
fNotesDatabase.child(noteID).removeValue().addOnComplet
eListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull Task<Void> task) {
          if (task.isSuccessful()) {
            Toast.makeText(NewNoteActivity.this, "Note
Deleted", Toast.LENGTH_SHORT).show();
            noteID = "no";
            finish();
          } else {
            Log.e("NewNoteActivity",
task.getException().toString());
            Toast.makeText(NewNoteActivity.this, "ERROR: "
+ task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
          }
        }
      });
    }
}
```

Note View Holder.java

```
package com.example.apprize;
import android.view.View;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;

public class NoteViewHolder extends
RecyclerView.ViewHolder {

    View mView;
    TextView textTitle,textTime;
    CardView noteCard;

    public NoteViewHolder(@NonNull View itemView) {

        super(itemView);
        mView = itemView;
        textTitle = mView.findViewById(R.id.note_title);
        textTime = mView.findViewById(R.id.note_time);
        noteCard = mView.findViewById(R.id.note_card);
    }

    public void setNoteTitle(String title){
        textTitle.setText(title);
    }

    public void setNoteTime(String time){
        textTime.setText(time);
    }
}
```

Note Model.java

```
package com.example.apprize;
```

```
public class NoteModel {

   public String noteTitle;
   public String noteTime;
   public NoteModel() {

   }
   public NoteModel (String noteTitle, String noteTime) {
      this.noteTitle = noteTitle;
      this.noteTime = noteTime;
   }

   public String getNoteTitle() {
      return noteTitle;
   }

   public void setNoteTitle(String noteTitle) {
      this.noteTitle = noteTitle;
   }

   public String getNoteTime() {
      return noteTime;
   }

   public void setNoteTime(String noteTime) {
      this.noteTime = noteTime;
   }
}
```

Get Time Ago.java

```
package com.example.apprize;
import android.content.Context;

public class GetTimeAgo {

   private static final int SECOND_MILLIS = 1000;
   private static final int MINUTE_MILLIS = 60 *
SECOND_MILLIS;
   private static final int HOUR_MILLIS = 60 *
MINUTE_MILLIS;
   private static final int DAY_MILLIS = 24 * HOUR_MILLIS;

   public static String getTimeAgo(long time, Context ctx){

      if (time < 1000000000000L) {

         // If timestamp given in seconds, convert to millis
         time *= 1000;
      }

      long now = System.currentTimeMillis();

      if (time > now || time <= 0){

         return null;
      }

      // TODO: localize
      final long diff = now - time;

      if (diff < MINUTE_MILLIS){
         return "just now";
      } else if (diff < 2 * MINUTE_MILLIS) {
         return "a minute ago";
      } else if (diff < 50 * MINUTE_MILLIS){
         return (diff / MINUTE_MILLIS + " minutes ago");
      } else if (diff < 90 * MINUTE_MILLIS) {
         return "an hour ago";
      } else if (diff < 24 * HOUR_MILLIS){
         return (diff / HOUR_MILLIS + " hours ago");
      } else if (diff < 48 * HOUR_MILLIS){
         return "yesterday";
      } else {
         return diff / DAY_MILLIS + " days ago";
      }
   }
}
```

Friends/Uplift Fragment

```
package com.example.apprize;
import android.os.Bundle;
import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.example.apprize.adapters.AdapterUsers;
import com.example.apprize.models.ModelUser;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.List;

public class FriendsFragment extends Fragment {

   RecyclerView recyclerView;
   AdapterUsers adapterUsers;
   List<ModelUser> userList;

   public FriendsFragment() {
      // Required empty public constructor
   }


   @Override
   public View onCreateView(LayoutInflater inflater,
ViewGroup container,
                            Bundle savedInstanceState) {
      // Inflate the layout for this fragment
      View view = inflater.inflate(R.layout.fragment_friends,
container, false);

      //init recycler view
```

```
        recyclerView =
view.findViewById(R.id.users_recyclerView);

        //set its properties
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new
LinearLayoutManager(getActivity()));

        //init user list
        userList = new ArrayList<>();

        //getAll users
        getAllUsers();
        return view;
    }

    private void getAllUsers() {

        //get current user
        final FirebaseUser fUser =
FirebaseAuth.getInstance().getCurrentUser();

        //get path of database named "Users" containing users
info
        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference("Users");

        //get all data from path
        ref.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {

                userList.clear();
                for (DataSnapshot ds: dataSnapshot.getChildren()) {

                    ModelUser modelUser =
ds.getValue(ModelUser.class);

                    //get all users except currently signed in user
                    if (!modelUser.getUid().equals(fUser.getUid())){

                        userList.add(modelUser);
                    }

                    //adapter
                    adapterUsers = new
AdapterUsers(getActivity(),userList);

                    //set adapter to recyclerview
                    recyclerView.setAdapter(adapterUsers);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {

            }
        });
    }
}
```

Chat Activity

```
package com.example.apprize;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import com.example.apprize.adapters.AdapterChat;
import com.example.apprize.models.ModelChat;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import com.squareup.picasso.Picasso;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class ChatActivity extends AppCompatActivity {

    //views from xml
    RecyclerView recyclerView;
    ImageView profileIv;
    TextView nameTv, userStatusTv;
    EditText messageEt;
    ImageButton sendBtn;

    //firebase auth
    FirebaseAuth firebaseAuth;
    FirebaseDatabase firebaseDatabase;
    DatabaseReference userDbRef;

    //for checking if user has seen or not
    ValueEventListener seenListener;
    DatabaseReference userRefForSeen;
    List<ModelChat> chatList;
    AdapterChat adapterChat;
    String hisUid;
    String myUid;
    String hisImage;

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_chat);

        //init views
        Toolbar toolbar  = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setTitle("");
        recyclerView = findViewById(R.id.chat_recyclerView);
        profileIv = findViewById(R.id.profileIv);
        nameTv = findViewById(R.id.userNameTv);
        userStatusTv = findViewById(R.id.userStatusTv);
        messageEt = findViewById(R.id.messageEt);
        sendBtn = findViewById(R.id.sendBtn);

        //Layout for RecyclerView
        LinearLayoutManager linearLayoutManager = new
LinearLayoutManager(this);
        linearLayoutManager.setStackFromEnd(true);

        //recycler View properties
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(linearLayoutManager);
        Intent intent = getIntent();
        hisUid = intent.getStringExtra("hisUid");
        firebaseAuth = FirebaseAuth.getInstance();
        firebaseDatabase = FirebaseDatabase.getInstance();
        userDbRef = firebaseDatabase.getReference("Users");

        //search user to get that user's info
        Query userQuery =
userDbRef.orderByChild("uid").equalTo(hisUid);

        //get user picture and name
        userQuery.addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {

                //check until required info is received
                for (DataSnapshot ds: dataSnapshot.getChildren()) {

                    //get data
                    String name = ""+ ds.child("name").getValue();
                    hisImage = ""+ ds.child("image").getValue();

                    //set data
                    nameTv.setText(name);
                    try {
                        //image received, set it to imageView in
toolbar

Picasso.get().load(hisImage).placeholder(R.drawable.ic_defa
ult_img_white).into(profileIv);
                    }
                    catch (Exception e) {
                        //there is exception getting picture, set default
picture

Picasso.get().load(R.drawable.ic_default_img_white).into(pr
ofileIv);
                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError
databaseError) {

            }
        });

        //click button to send message
        sendBtn.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {

                //get text from edit text
                String message =
messageEt.getText().toString().trim();

                //check if text is empty
                if (TextUtils.isEmpty(message)){
                    //text empty
                    Toast.makeText(ChatActivity.this, "Cannot send
empty message...", Toast.LENGTH_SHORT).show();
                }
                else {
                    //text not empty
                    sendMessage(message);
                }
            }
        });
        readMessages();
        seenMessages();
    }

    private void seenMessages() {

        userRefForSeen =
FirebaseDatabase.getInstance().getReference("Chats");
        seenListener =
userRefForSeen.addValueEventListener(new
ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {

                for (DataSnapshot ds: dataSnapshot.getChildren()){

                    ModelChat chat = ds.getValue(ModelChat.class);

                    if (chat.getReceiver().equals(myUid) &&
chat.getSender().equals(hisUid)){

                        HashMap<String, Object> hasSeenHashMap =
new HashMap<>();
                        hasSeenHashMap.put("isSeen", true);
```

```
                ds.getRef().updateChildren(hasSeenHashMap);
            }
          }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }
      });
  }

  private void readMessages() {

    chatList = new ArrayList<>();
    DatabaseReference dbRef =
FirebaseDatabase.getInstance().getReference("Chats");

    dbRef.addValueEventListener(new ValueEventListener()
{
        @Override
        public void onDataChange(@NonNull DataSnapshot
dataSnapshot) {

          chatList.clear();
          for (DataSnapshot ds: dataSnapshot.getChildren()){
            ModelChat chat = ds.getValue(ModelChat.class);
            if (chat.getReceiver().equals(myUid) &&
chat.getSender().equals(hisUid) ||
                chat.getReceiver().equals(hisUid) &&
chat.getSender().equals(myUid) ){
              chatList.add(chat);
            }
            //adapters
            adapterChat = new
AdapterChat(ChatActivity.this, chatList, hisImage);
            adapterChat.notifyDataSetChanged();
            //set adapter to recyclerview
            recyclerView.setAdapter(adapterChat);
          }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError
databaseError) {

        }
      });
  }

  private void sendMessage(String message) {

    DatabaseReference databaseReference =
FirebaseDatabase.getInstance().getReference();

    String timestamp =
String.valueOf(System.currentTimeMillis());

    HashMap<String, Object> hashMap = new
HashMap<>();
    hashMap.put("sender", myUid);
    hashMap.put("receiver", hisUid);
    hashMap.put("message", message);
    hashMap.put("timestamp", timestamp);
    hashMap.put("isSeen", false);

databaseReference.child("Chats").push().setValue(hashMap)
;

    //reset edittext after send
    messageEt.setText("");
  }

  private void checkUserStatus() {
    //get current user
    FirebaseUser user = firebaseAuth.getCurrentUser();
    if (user != null) {

      myUid = user.getUid();
    }
    else  {
      startActivity(new Intent(ChatActivity.this,
MainActivity.class));
      finish();
    }
  }

  @Override
  protected void onStart() {
    checkUserStatus();
    super.onStart();
  }

  @Override
  protected void onPause() {
    super.onPause();
    userRefForSeen.removeEventListener(seenListener);
  }

  @Override
  public boolean onCreateOptionsMenu(Menu menu) {

    getMenuInflater().inflate(R.menu.menu_main, menu);
    return super.onCreateOptionsMenu(menu);
  }

  @Override
  public boolean onOptionsItemSelected(@NonNull
MenuItem item) {

    int id = item.getItemId();

    if (id == R.id.action_logout){
          firebaseAuth.signOut();
      checkUserStatus();
    }
    return super.onOptionsItemSelected(item);
  }
}
```

Chat Adapter.java

```
package com.example.apprize.adapters;
import android.content.Context;
import android.os.Build;
import android.text.format.DateFormat;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.constraintlayout.widget.ConstraintLayout;
import androidx.recyclerview.widget.RecyclerView;
import com.example.apprize.R;
import com.example.apprize.models.ModelChat;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;
import org.w3c.dom.Text;
import java.util.Calendar;
import java.util.List;
import java.util.Locale;

public class AdapterChat extends
RecyclerView.Adapter<AdapterChat.MyHolder> {

   private static final int MSG_TYPE_LEFT = 0;
   private static final int MSG_TYPE_RIGHT = 0;
   Context context;
   List<ModelChat> chatList;
   String imageUrl;
   FirebaseUser fUser;

   public AdapterChat(Context context, List<ModelChat>
chatList, String imageUrl) {
      this.context = context;
      this.chatList = chatList;
      this.imageUrl = imageUrl;
   }

   @NonNull
   @Override
   public MyHolder onCreateViewHolder(@NonNull
ViewGroup viewGroup, int i) {

      //inflate layouts: row_chat_left.xml for receiver,
row_chat_right.xml for sender
      if (i == MSG_TYPE_RIGHT){
         View view =
LayoutInflater.from(context).inflate(R.layout.row_chat_right
, viewGroup, false);
         return new MyHolder(view);
      }
      else {
         View view =
LayoutInflater.from(context).inflate(R.layout.row_chat_left,
viewGroup, false);
         return new MyHolder(view);
      }
   }

   @RequiresApi(api = Build.VERSION_CODES.N)
   @Override
   public void onBindViewHolder(@NonNull MyHolder
myHolder, int i) {

      //get data
      String message = chatList.get(i).getMessage();
      String timeStamp = chatList.get(i).getTimestamp();

      //convert timestamp to dd/mm/yyyy hh:mm am/pm
      Calendar cal = Calendar.getInstance(Locale.ENGLISH);
      cal.setTimeInMillis(Long.parseLong(timeStamp));
      String dateTime = DateFormat.format("dd/MM/yyyy
hh:mm aa", cal).toString();

      //set data
      myHolder.messageTv.setText(message);
      myHolder.timeTv.setText(dateTime);
      try {
         Picasso.get().load(imageUrl).into(myHolder.profileIv);
      }
      catch (Exception e) {
      }

      //set seen/delivered
      if (i == chatList.size()-1) {
         if (chatList.get(i).isSeen()) {
            myHolder.isSeenTv.setText("Seen");
         }
         else {
            myHolder.isSeenTv.setText("Delivered");
         }
      }
      else {
         myHolder.isSeenTv.setVisibility(View.GONE);
      }
   }

   @Override
   public int getItemCount() {
      return chatList.size();
   }

   @Override
   public int getItemViewType(int position) {
      //get currently signed in user
      fUser = FirebaseAuth.getInstance().getCurrentUser();
      if
(chatList.get(position).getSender().equals(fUser.getUid())) {

         return MSG_TYPE_RIGHT;
      }
      else {
         return MSG_TYPE_LEFT;
      }
   }

   //view holder class
   class MyHolder extends RecyclerView.ViewHolder{
```

```
    //views
    ImageView profileIv;
    TextView messageTv, timeTv, isSeenTv;

    public MyHolder(@NonNull View itemView) {
      super(itemView);

      //init views
      profileIv = itemView.findViewById(R.id.profileIv);
      messageTv =
itemView.findViewById(R.id.rcmessageTv);
      timeTv = itemView.findViewById(R.id.rctimeTv);
      isSeenTv = itemView.findViewById(R.id.isSeenTv);
    }
  }
}
```

Users Adapter.java

```
package com.example.apprize.adapters;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.example.apprize.ChatActivity;
import com.example.apprize.R;
import com.example.apprize.models.ModelUser;
import com.squareup.picasso.Picasso;
import java.util.List;

public class AdapterUsers extends
RecyclerView.Adapter<AdapterUsers.MyHolder>{

  Context context;
  List<ModelUser> userList;

  //constructor
  public AdapterUsers(Context context, List<ModelUser>
userList) {
    this.context = context;
    this.userList = userList;
  }

  @NonNull
  @Override
  public MyHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {

    //inflate layout (row_user.xml)
      View view =
LayoutInflater.from(context).inflate(R.layout.row_users,
parent, false);

    return new MyHolder(view);
  }

  @Override
  public void onBindViewHolder(@NonNull MyHolder
holder, int i) {

    //get data
    final String hisUID = userList.get(i).getUid();
    String userImage = userList.get(i).getImage();
    String userName = userList.get(i).getName();
    final String userEmail = userList.get(i).getEmail();

    //set data
    holder.mNameTv.setText(userName);
    holder.mEmailTv.setText(userEmail);
    try {


Picasso.get().load(userImage).placeholder(R.drawable.ic_fac
e_white).into(holder.mAvatarIv);
    }
    catch (Exception e) {
    }

    //handle item click
    holder.itemView.setOnClickListener(new
View.OnClickListener() {

      @Override
      public void onClick(View view) {

        Intent intent = new Intent(context,
ChatActivity.class);
        intent.putExtra("hisUid", hisUID);
        context.startActivity(intent);
      }
    });
  }

  @Override
  public int getItemCount() {
    return userList.size();
  }

  //view holder class
  class MyHolder extends RecyclerView.ViewHolder{

    ImageView mAvatarIv;
    TextView mNameTv, mEmailTv;

    public MyHolder(@NonNull View itemView) {

      super(itemView);
      mAvatarIv =
itemView.findViewById(R.id.avatarCircularIv);
      mNameTv =
itemView.findViewById(R.id.name_Tv_Tv);
      mEmailTv =
itemView.findViewById(R.id.email_Tv_Tv);
    }
  }
```

```
}
```

Chat Model.java

```
package com.example.apprize.models;

public class ModelChat {
   String message, receiver, sender, timestamp;
   boolean isSeen;

   public ModelChat () {

   }

   public ModelChat(String message, String receiver, String
sender, String timestamp,
                    boolean isSeen) {
      this.message = message;
      this.receiver = receiver;
      this.sender = sender;
      this.timestamp = timestamp;
      this.isSeen = isSeen;
   }

   public String getMessage() {
      return message;
   }

   public void setMessage(String message) {
      this.message = message;
   }

   public String getReceiver() {
      return receiver;
   }

   public void setReceiver(String receiver) {
      this.receiver = receiver;
   }

   public String getSender() {
      return sender;
   }

   public void setSender(String sender) {
      this.sender = sender;
   }

   public String getTimestamp() {
      return timestamp;
   }

   public void setTimestamp(String timestamp) {
      this.timestamp = timestamp;
   }

   public boolean isSeen() {
      return isSeen;
   }

   public void setSeen(boolean seen) {
      isSeen = seen;
   }
}
```

User Model.java

```
package com.example.apprize.models;

public class ModelUser {

   //use same name as in firebase database
   String name, email, search, quote, image, cover, uid;

   public ModelUser() {

   }

   public ModelUser(String name, String email, String search,
String quote, String image,
                    String cover, String uid) {
      this.name = name;
      this.email = email;
      this.search = search;
      this.quote = quote;
      this.image = image;
      this.cover = cover;
      this.uid = uid;
   }

   public String getName() {
      return name;
   }

   public void setName(String name) {
      this.name = name;
   }

   public String getEmail() {
      return email;
   }

   public void setEmail(String email) {
      this.email = email;
   }

   public String getSearch() {
      return search;
   }

   public void setSearch(String search) {
      this.search = search;
   }

   public String getQuote() {
      return quote;
   }

   public void setQuote(String quote) {
      this.quote = quote;
   }
```

```
public String getImage() {
      return image;
   }

   public void setImage(String image) {
      this.image = image;
   }

   public String getCover() {
      return cover;
   }

   public void setCover(String cover) {
     this.cover = cover;
   }

   public String getUid() {
      return uid;
   }

   public void setUid(String uid) {
      this.uid = uid;
   }
}
```

User's Manual

Phone Requirements

- The required mobile specification for the mobile application to be installed is Android OS version Lollipop (5.0) up to the latest version.
- The Android phone must at least have 1GB of RAM and internal or external memory.
- Apprize can't be utilized offline and was also developed for the Android Platform only

Instructions for downloading the application from Google Play Store.

- Open Google Play Store in your mobile phone.
- Search for Apprize: A Social Platform for Self-Wellness Using Decision Tree Algorithm.
- Tap Install.

Steps on How to Use.

- The end-user must register an Apprize account or login using a Google account.
- After a successful registration, the mobile application will prompt a quick Personality Test to be answered.
- The assessment will define the end-users' MBTI personality type that will be displayed at the Profile tab.
- The Apprize user can navigate the mobile application' feature from the Profile Tab including all the additional information or to enter new data.
- Tap the Apprize Tab on the navigation bar of the application to see the Apprize activities or features included in the mobile application.
- Select the Uplift Tab on the navigation bar to open the Social Platform feature.
- Once the user chooses another Apprize user on the Social Platform, they may interact with each other through the messaging feature of the mobile application.

John Christopher L. Austria

Mabini, Purok 2 Lipa City, Batangas

+63 926 625 2662

johnchristopheraustria@lpubatangas.edu.ph

PERSONAL INFORMATION

Age	:	23
Date of Birth	:	April 9, 1997
Gender	:	Male
Civil Status	:	Single
Height	:	5’5”
Weight	:	40kg
Nationality	:	Filipino-American
Religion	:	Roman Catholic

EDUCATIONAL BACKGROUND

Tertiary: **Lyceum of the Philippines University Batangas**
Bachelor of Science in Computer Science
Capitol Site, Batangas City
2019 - present

Secondary: **Sto. Niño Formation and Science School**
J. Belen St. Rosario, Batangas
2009-2013

Elementary: **Sto. Niño Formation and Science School**
J. Belen St. Rosario, Batangas
2007-2009

Jazril Karlo P. Bagui

National Hi-way, Alangilan, Batangas City

+63 947 397 9601

jazrilbagui@lpubatangas.edu.ph

PERSONAL INFORMATION

Age	:	22
Date of Birth	:	July 24, 1998
Gender	:	Male
Civil Status	:	Single
Height	:	5’2”
Weight	:	53kg
Nationality	:	Filipino
Religion	:	Roman Catholic

EDUCATIONAL BACKGROUND

Tertiary: **Lyceum of the Philippines University Batangas**
Bachelor of Science in Computer Science
Capitol Site, Batangas City
2015 – 2020

Secondary: **Immaculate Heart of Mary Learning Center & School of Values**
Kumintang Ilaya, Batangas City
2011 – 2015

Elementary: **Immaculate Heart of Mary Learning Center & School of Values**
Kumintang Ilaya, Batangas City
2005 – 2011

Dalisay, Mark Kenneth B.

Kumintang Ilaya, Batangas City

+63 916 404 6730

kennethdalisay@lpubatangas.edu.ph

PERSONAL INFORMATION

Age	:	24
Date of Birth	:	July 2, 1996
Gender	:	Male
Civil Status	:	Single
Height	:	5’8”
Weight	:	50kg
Nationality	:	Filipino
Religion	:	Roman Catholic

EDUCATIONAL BACKGROUND

Tertiary: **Lyceum of the Philippines University Batangas**
Bachelor of Science in Computer Science
Capitol Site, Batangas City
2016 – Present

Secondary: **Batangas State University**
Rizal Avenue Extension, Batangas City
2008 – 2012

Elementary: **Casa del Bambino Emmanuel Montessori**
Contreras Compound, Alangilan, Batangas City
2002 – 2008

Nick Paolo Lodana

Lian, Batangas.

+63 920 691 7716

nickpaololodana@lpubatangas.edu.ph

PERSONAL INFORMATION

Age	:	23
Date of Birth	:	April 1, 1997
Gender	:	Male
Civil Status	:	Single
Height	:	5’9
Weight	:	50kg
Nationality	:	Filipino
Religion	:	Born Again Christian

EDUCATIONAL BACKGROUND

Tertiary: **Lyceum of the Philippines University Batangas.**
Bachelor of Science in Computer Science
Capitol Site, Batangas City
2014- Present

Secondary: **Lian Institute.**
Lian, Batangas
2010-2014

Elementary: **Lian Central School.**
Lian, Batangas
2004-2010

2nd Result - Apprize A Social Platform for Self-Wellness Using Decision Tree Algorithm

ORIGINALITY REPORT

9%	4%	0%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.truity.com Internet Source	3%
2	Submitted to Christ University Student Paper	2%
3	Submitted to Kensington College of Business Student Paper	1%
4	Submitted to Mount Si High School Student Paper	1%
5	Submitted to Mississippi Valley State University Student Paper	<1%
6	Submitted to Universiti Teknologi MARA Student Paper	<1%
7	www.kingaafrica.org Internet Source	<1%
8	themindsjournal.com Internet Source	<1%
9	Submitted to Colorado State University, Global	

	Campus Student Paper	<1%
10	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1%
11	docplayer.net Internet Source	<1%
12	pergamos.lib.uoa.gr Internet Source	<1%
13	www.ncbi.nlm.nih.gov Internet Source	<1%
14	www.semanticscholar.org Internet Source	<1%