**KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**COLLEGE OF SCIENCE**

**FACULTY OF PHYSICAL AND COMPUTATIONAL SCIENCES**

**DEPARTMENT OF COMPUTER SCIENCE**

**Dr. KATE TAKYI – SUPERVISOR**

**MINI-PROJECT**



**COUNSELING MOBILE APPLICATION**

**BY**

**KOOMSON PRINCE KWAKU**

**4213420**

**20748544**

Table of Contents

[CHAPTER 1 1](#_Toc141714133)

[**Introduction** 1](#_Toc141714134)

[1.1 Background of Study 1](#_Toc141714135)

[1.2 Problem Statement 1](#_Toc141714136)

[1.3 Aim of the project 2](#_Toc141714137)

[1.4 Specific Objectives of the project 2](#_Toc141714138)

[1.5 Justification of project 3](#_Toc141714139)

[1.6 Motivation for undertaking Project 3](#_Toc141714140)

[1.7 Scope of project 4](#_Toc141714141)

[1.8 Project limitations 6](#_Toc141714142)

[1.9 Beneficiaries of the project 6](#_Toc141714143)

[1.10 Academic and practical relevance of the project 6](#_Toc141714144)

[1.11 Structure of report 7](#_Toc141714145)

[1.12 Project Deliverables 8](#_Toc141714146)

[CHAPTER 2 10](#_Toc141714147)

[Review of related works / Review of similar systems 10](#_Toc141714148)

[2.1 Processes of the existing system. 10](#_Toc141714149)

[2.2 The proposed system 12](#_Toc141714150)

[2.3Conceptual Design 13](#_Toc141714151)

[2.4 Architecture of the proposed system. 14](#_Toc141714152)

[2.5 Components Designs and Components descriptions. 14](#_Toc141714153)

[2.6 Proposed system/software features. 14](#_Toc141714154)

[2.7 Development tools and environment 16](#_Toc141714155)

[2.8 Benefits of implementation of the proposed system. 16](#_Toc141714156)

[CHAPTER 3 18](#_Toc141714157)

[**Methodology** 18](#_Toc141714158)

[3.1Chapter Overview 18](#_Toc141714159)

[3.2 Requirement specification 18](#_Toc141714160)

[3.3 Stake holders of system 18](#_Toc141714161)

[3.4 Requirement Gathering process 19](#_Toc141714162)

[3.5 Functional requirements 19](#_Toc141714163)

[3.6 UML Diagrams 20](#_Toc141714164)

[USE CASE DESCRIPTION 21](#_Toc141714165)

[3.7 Non – functional requirements 25](#_Toc141714166)

[3.8 Security concepts 25](#_Toc141714167)

[3.9 Project methods (either agile or plan driven) 26](#_Toc141714168)

[3.10 Chosen model and justification. 26](#_Toc141714169)

[3.11 PROJECT DESIGN CONSIDERATION (LOGICAL DESIGNS) 26](#_Toc141714170)

[UI Design 26](#_Toc141714171)

[3.12 DB Design 28](#_Toc141714172)

[3.13 Developmental tools 29](#_Toc141714174)

[CHAPTER 4 30](#_Toc141714175)

[IMPLEMENTATION AND RESULTS 30](#_Toc141714176)

[4.1 Chapter Overview. 30](#_Toc141714177)

[4.2 Mapping logical design onto physical platform 31](#_Toc141714178)

[algorithm for implementing UI 31](#_Toc141714179)

[algorithm for implementation of database 66](#_Toc141714181)

[4.3 Testing 68](#_Toc141714182)

[4.4 Results 68](#_Toc141714183)

[CHAPTER 5 69](#_Toc141714184)

[FINDINGS AND CONCLUSION 69](#_Toc141714185)

[5.1 Chapter Overview 69](#_Toc141714186)

[5.2 Findings 69](#_Toc141714187)

[5.3 Conclusions 69](#_Toc141714188)

[5.4 Challenges/ limitations of the system 69](#_Toc141714189)

[5.5 Lesson learnt 70](#_Toc141714190)

[5.6 Recommendations for future works 70](#_Toc141714191)

[5.7 Reference 70](#_Toc141714192)

# CHAPTER 1

# **Introduction**

## Background of Study

In today’s fast growing and interconnected world, the engage of individuals have become increasingly intense, bringing a strain on our mental health and emotional well-being. With the rise in emotional anxiety, stress from our daily lives, and other mental health challenges, there is a need for easily accessible and effective counseling services that can help and assist individuals seeking for help and assistance. The counseling app will be a comprehensive and logical solution that embraces the ideas of confidentiality, empath y and personalized care. Understanding the modern challenges of balancing academic life, personal and social life in a new environment on campus is a lot at times confusing not only to freshers but continuing students as well.

Counselling app offers flexible schedule options where users can book, chat or call to make an appointment at the comfort of their residence, eliminating the need for physical visits and ensuring privacy and comfort. To further enhance this experience, the counselling app incorporates interactive tool and resources which users can actively use in their journey towards mental and emotional well-being.

With the counseling app, we aspire to make counseling services accessible to a broader audience, breaking down geographical barriers and providing to individual from around campus. And together, we can ensure the mental wellness, fostering resilience and empowering users to lead fulfilling lives.

## Problem Statement

Students, especially freshers are accustomed to leaning on peers, newly found friends and senior course mates concerning issues about their newly found living environment, forgetting the fact that the school has the KNUST COUNSELLING CENTER (KCC) ready and at their beckoning call to assist them with the full might of the school, to make their campus life as blissful as humanly possible. Parents tend to feel uneasy about the kind of life their children are living on campus and whether the right supervision and care is being put in place to ensure the welfare of their children. A school able to achieve such a feat, acquires the full trust and assurance of the parents of student in the school, which goes a mile to raise the general standard of the school.

My thorough research led me to discover that majority of students have no idea the location, aim and function of the KNUST COUNSELLING CENTER, even with the countless effort placed down by the center to engage students in all their events and news. This makes the job of the counsel reaching out and fulfilling their duties to the students tedious.

Hence the initiative to build an online app which seeks to help the counseling center to engage students in their daily activities and help solve the issues or problems of students, being most at times mislead with information and advices from fellow students which can be easily, strategically and purposely done by the counseling center. The app seeks to show the importance of the relationship between the counseling center and the KNUST student populace and the relevance of the counseling center on campus and in the academic life of students at large. The app will come with well thought of key elements, which seeks to display events and news, provide a platform for students to interact with counselors, call counselors in times of need, listen to audio messages approved by the counseling sector and student chaplaincy, and book appointments with scheduled dates with well trained counselors to help students with their daily situations on campus.

## Aim of the project

The aims are the required and specific goals set forth to be accomplished by the mobile application, to help give reason behind its development. The mobile app serves as a means of reaching out to the entire KNUST school populace, for them to be aware of the importance and relevance of the counselling center in their academic and day-to-day lives throughout their stay in KNUST. It also aims to give the students a mind of comfort and assurance, knowing that there are parent and guardian figures ready to help them and cater for their needs as much as humanly possible. The app is meant to also give parents of students, the peace of mind gotten through knowing perfectly well that your ward is being carefully monitored, assisted and given needed time for their well-being. The mobile app will aim to drive the purpose and ambitions of the counselling sector home. Provide an avenue where the counselling sector could express their view and make upcoming events public to the whole KNUST populace.

## Specific Objectives of the project

* My plan is to build a comprehensive, easy to use and relatable mobile application, using flutter and dart programming language that takes and relays the situations of students and provide credible and reliable feedback from well-known, accredited, accepted and respected counselors across the breadth and length of KNUST COUNSELLING CENTER.

* The mobile app should bridge the gap between students and the KNUST COUNSELLING CENTER, ensuring a continuous flow of harmony which tends to strengthen the relationship each student will need in their daily conversations with their appointed counsellors.
* The app is to give students an opportunity and act as a gate way to help them locate the needed help their parents or guardians won’t be able to provide due to the physical distance between them.
* The app should make the motives and plans of the counselling center clear and precise toe the entire populace without any ambiguity.
* The app should serve as a platform where students get the latest update and news about ins and outs of programs happening or being organized by the counselling center

## Justification of project

My initiative will be necessary to alleviate the stress students go through relying on information from their fellow peers. The app will ensure that students make the required and suitable choice based on the verified and trusted information they receive from the counsellors.

The app will also seek to drastically reduce the immense effort put in by the KNUST COUNSELLING CENTER in gaining the attention of students through notice board and other social media outlets which yields less acknowledgement by the student populace. This effort will be reduced by the news and events sections found on the homepage and notifications received by students.

The app will help freshers who arrive on campus extensively. By helping them connect deeply with the school authorities in the form of KCC to help them settle in comfortable in their new found environment instead of asking fellows students and senior course mates who may not be well suited in giving advice and assistance duly needed.

It will provide an avenue where students can contact for help and assistance in times of need, stress, depression, hardship, etc. and provide them the needed aid from well trained and accredited counsellors across the length and breadth of the KNUST campus. Ensuring that students are provided with the needed help and assistance in their academics and campus lives to peaceful academic tenure for the student.

The general awareness of the app will create a sense of ease and peace in the minds of parents knowing perfectly well that their wards are in constant interaction with qualified school authorities and that the safe and well-being of their children are secured.

## Motivation for undertaking Project

The KNUST COUNSELLING CENTER has made many strides and effort aimed at providing quality emotional support to the student populace to ensure a smooth stay on campus. These efforts were well executed and had immense impact on the lives of students on campus. Though the success of the initiative had the small number of students being admitted at that period being a major factor, we can clearly see that those efforts coupled with newly established ones are having little to no effect in reaching majority of the targeted audience. This very concept was a starting ignition which light a desire in me to help assist in making the lives of students on campus a as stress free as possible using **ICARE.**

To help students struggling with situations of depression, which can arrive from many factors in an individual’s life. Depression is a fatal emotional stress which has been taken lightly due the nature of social life in Ghana, were we hardly notice the struggles of our fellow people due to our busy economy and the mindset that we live in a fun, lively and peaceful country, this makes it difficult to notice or know how to help out people struggling with it. Therefore, I felt the edge to use an innovative and modern form in the frame of and app which provides students a safe haven and a place to seek the needed assistance and guidance from professional who have year under their belts in dealing with cases involving and revolving around depression.

Help reduce the over reliance of students on their friends who have no experience in dealing with problems of emotional well-being. Majority of students, when overcome with any form of emotional pain or stress tend to see their closest friends as their initial point of interaction to help them find answers in their times of need. But these friends may have close to no experience in dealing with many problems their friends bring their way, and this leads them to giving advices which are not needed or suited for that particular situation. With the establishment of **ICARE,** students will have a place to go and seek appropriate help from professionals who have had enough times dealing with situations similar to theirs and trained peer counsellors who also extend same services.

Bridging the gap existing between the student populace and the KNUST COUNSELLING CENTER. Students seem less interested in the activities being put in place by the center in bringing them closer, so students could confide in them more freely and openly at all times. Although many efforts have been put in place such as the establishment of peer counselors and organizing various event to interact with students and teach them how to deal with various emotional stress encountered as a student, have been put in place the number of targets reached is not really promising. Which also led me to build the app, so that instead of students moving to search for the needed solution for their problem from friends or feeling shy to approach a counsellor. They can easily get the needed information from the comfort of their hostel rooms or halls.

## Scope of project

The project will serve as a form of a bridge “metaphorically speaking” that will strive to bring both the student’s populace and the counseling sector together in terms of communication and bring to the comfort of the student homes all the necessary information and events purposed for the students. The app is meant to provide the students of KNUST with a safe haven where they can freely and comfortably seek help and advice concerning any form of emotional problem which is geared toward their campus lives, academic lives and any other social crisis they may be facing at the moment. It is meant to provide students with information concerning the goals and ideals that drives the counseling sector in seeking the very best for the students. It will also guide newly admitted students (Freshers) with a place to seek advice and help, to help them adjust well and comfortably in their newly found environment. This will help curb freshers from indulging in dangerous or unneeded acts which will go a long way in affecting their stay here or tarnishing their future and helping bring about a safe and peaceful campus life style. The project will serve as a platform that will help the counselling center by giving them an avenue to post upcoming events and news updates to help students stay in tune with what is happening on campus and when and where events that seek to help them in terms of their emotional well-being are going to be held.

The project is to give a sense of satisfaction to a student in distress and a place where they can easily seek help and be provided with suitable reply or answers in the form of audios, booking of appointments, calling the counselling center or interacting and chatting a counsellor. Provide an avenue where students can listen to productive and insightful audio messages provided and approved by the counselling center pertaining to their emotional well-being and care. The project will provide a chat platform for students to reach out to both counselors and peer counselors in times of need and emotional problems. It has a booking page which assists students in making appointments with their various college counsellors to meet them and have a personal talk with them, concerning any situations they might be going through. It also uses a homepage which has an event and news list view, which provides the user with the latest information concerning the activities of the center.

The app will require the full acknowledgement of the KNUST administration, the chaplaincy, the KNUST COUNSELING CENTER and the entire student populace to ensure the effective utilization of the mobile app and it features. The app will encompass the main KNUST campus and it affiliated offsite campuses such as the obuasi campus using the school email address, index numbers and reference numbers as the authentication base.

## Project limitations

* The app will be limited due to network issues based on the geographical location of the students. Due to the undeniable fact that students would need internet bundle to access the app, bad connections or low and no internet bundle time will make accessing the app impossible.
* Aside students, lectures who seek to use the app for their benefits will not be allowed to access the app.
* Graduates who have completed their term of enrollment in the school and also teaching assistance will not be permitted to use the app.
* Unable to conduct video calls with counselor.
* Delay in message replies due to the busy scheduled of the counselors and peer counselors.

## Beneficiaries of the project

The System will serve to benefit both the KNUST COUNSELING CENTOR and the entire student populace of the KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY. Where there counseling center will update students with upcoming events and new and the students also will be able to reach out and interact with the counseling center to help cater for their emotional well-being. It will help reduce the immense rise in depression, academic stress and emotional difficulties felt by students.

The project will also serve the school administration in terms of ensuring both the mental and emotional well-being of the students are being taken care of in their times of distress and need. Ensuring a well lubricated and functional system of management without fear of students endangering themselves.

## Academic and practical relevance of the project

The System comes with a lot of academic and practical importance which include but not restricted to:

* Students feeling overwhelmed by the immense pressure and strain from academics will be able to call a counselor, book and appointment or listen to approved audio messages at the comfort of their abode and at any specified time.
* Students, who are at a loss as to how to cope with their social life style and academics will be able to get notified of any upcoming academic talk from the event section on the homepage and be able to chat with counselors’ concerning any form of difficulties they may face.
* Students suffering from cases of depression, due to their academic welfare, financial problems or any social hindrance will find it easier reaching out to the counseling center through the app than to go out confiding in a friend.
* Freshers who have academic scares for their very first exams on campus will have a place to unburden themselves and seek advice as to how to overcome it.

## Structure of report

The report or documentation has been structured into five main chapters which cover the entirety of the developmental process of the mobile application. The report begins with Chapter one which seeks to introduce the main relevance and brief insight into the project on it aims and purposes. Chapter one covers and elaborates on various topics such as:

* The problem statement, which defines the aim and motive behind the development of the project and why it’s of any relevance.
* The aims and objects that this project seek to thrive on and in a long run accomplish them. The aims and objectives set the standard and goal the app seeks to achieve.
* Justification and motivation for conforming or accepting to perform the particular project.
* Scope of the project that defines the boundaries, purpose of the project, key individuals involved in the project and the limitations associated with the project.
* The beneficiaries of the project
* The academic and practical relevance of the project, in the life of the customers.
* The planning, time allocation and the activities done to acquire or placed into the development and acquisition of the project in its current state.
* Finally, the deliverables or what is to be expected of a functioning project and it benefits to the stakeholders and customers.

The chapter two which is the literary review of related work seek, tries to see reason and importance in project chosen and it relativity any already developed and existing projects. The chapter two elaborates more firmly on:

* Process the existing of any existing system, detailing its system features, advantages and disadvantages and disadvantage of the existing related system.
* The proposed system, establishing it; dominance over the existing and already in use system.
* The conceptual frame work, which is the initial framework of how the system is to look like in the near future.
* The architecture of proposed system, that seeks to elaborate on the internal happening of the project or how the operates internally.
* Components design and components description. These aspects elaborate on the on the various functions of the components in the architectural design.
* Proposed system features that differentiates it from its previous predecessors.
* The developmental tool and environment in which the project was built.
* Benefits of implementing or using the proposed system

Chapter three talks about the various methodology utilized in the development of the project and the needs and requirements put forward by the both the stakeholders and consumers and various diagrams depicting and describing the functionality of the system giving a more pictorial look to the project in its developmental stage. Chapter three contains sub topics such as:

* Chapter overview, which gives a brief introduction into the content in the system.
* Requirement specifications handed down by the stakeholders.
* Highlights the various stakeholders involved in the development of the system.
* The process and strategies used in the gathering process of the various information and solutions relating to the requirements.
* Functional requirements involved or associated with the project.
* The UML Diagrams and its Description.
* Non-functional requirements.
* Security concepts defining the protection layers around the app.
* The project method used in the development of the app.
* Various software process models.
* Chosen modal and its justifications.
* The logical design of the project UI and DB design including E-R Diagram and SCHEMA.
* Developmental tools and how it was used in the development of the project.

Implementation placed down to achieve in the project and the various results obtained from then Project are defined in chapter four of this documentation. It defines and shows :

* Chapter overview, which elaborate on the various subtopics under the it.
* Mapping logical design onto a physical platform
* Testing
* Testing plan
* Results.

The last chapter details the findings and conclusion of the project stating:

* Chapter overview
* The findings
* Conclusion
* Challenges
* Lessons learnt
* Recommendations for future works
* Recommendation for project commercialization.
* Reference

## Project Deliverables

The healthcare app will require a student’s accurate credentials which will be cross checked with the backend database to give the student access to the app. This will ensure safety and secure a student’s privacy on the app. The authenticity of the student’s username or school e-mail address, index number and password will grant them access.

The homepage will produce as an output, events which will give students the fore knowledge of an activity KCC has placed in the pipeline for their sakes and helps generate interest in these events. It will also display news which will inform students about their achievements and efforts they are putting in to ensure a smooth end efficient environment for students.

The app will contain an audio page which will contain provide students with a range of carefully selected audio messages provided by both KCC and the student counsel to help ensure the growth of the student both mentally and spiritually.

The health care app will provide students with feedback from counsellors on the chat page to ensure effective communication and response time. It will also provide confirmation on bookings sent to counsellors through the booking page. Ensuring that student have a definite time and location for the appointments.

# CHAPTER 2

# Review of related works / Review of similar systems

## 2.1 Processes of the existing system.

With increasing rate of technological advancement, many counseling apps have been integrated into the system but each of these systems have their pros and cons. This systems include Replika, BetterHelp and TalkSpace.

Replika is an AI-powered chatbot designed to engage users in meaningful conversations and act as a personal AI companion. Developed by OpenAI. Replika utilizes natural language processing and machine learning algorithms to learn from user interactions and simulate human-like responses. The primary goal of Replika is to provide users with a supportive and non-judgmental space for self-reflection, personal growth, and companionship.

Pros:

* 24/7 Availability: Replika is accessible at any time, allowing users to engage in conversations whenever they feel the need for companionship or support. This continuous availability can be particularly beneficial for those who may feel isolated or lonely.
* Non-Judgmental Listener: Replika is a non-judgmental AI companion that offers a safe space for users to share their thoughts and feelings without fear of criticism or rejection. This aspect can be comforting for individuals who may find it challenging to open up to others.
* Personalized Conversations: As users interact with Replika, the AI learns from these interactions and tailors its responses to suit individual preferences and communication styles. This personalization creates a more engaging and human-like conversation experience.
* Therapeutic Value: Replika can act as a form of therapeutic intervention for some users. Engaging in conversations with an AI can help users explore their emotions, thoughts, and experiences, fostering self-awareness and self-reflection.

Cons:

* Limitations of AI Companionship: While Replika is designed to simulate human-like conversations, it is still an artificial intelligence and lacks true emotional understanding. It may not fully replace the depth of human companionship and empathy that real-life interactions can provide.
* Repetitive Responses: Replika's AI has limitations in terms of generating novel and varied responses, especially when users engage in extended or complex conversations. This can make interactions feel repetitive and less engaging over time.
* Lack of Emotional Context: Replika may struggle to understand emotional nuances or provide appropriate responses to complex emotional situations. Users seeking emotional support for significant life challenges may find the AI's responses inadequate.
* Dependency Risks: Some users may develop an emotional attachment to their Replika, leading to potential dependency on an AI for companionship and emotional support. Relying solely on an AI for human-like interactions can hinder social skills development.

Betterhelp is one of the leading online counseling platforms that offers accessible and professional mental health support through virtual therapy sessions. Launched in 2013, Betterhelp provides users with a network of licensed therapists and counselors who offer personalized counseling services via text, chat, phone, or video sessions. The platform aims to make therapy more convenient and flexible, catering to individuals with various mental health needs.

Pros:

* Accessibility and Convenience: Betterhelp enables users to access counseling services from the comfort of their homes or any location with internet access. This convenience eliminates the need for physical visits to a therapist's office, making therapy more accessible to those with mobility issues or living in remote areas.
* Wide Range of Therapist Specializations: The platform offers a diverse network of licensed therapists and counselors with expertise in various areas of mental health. Users can find a professional who aligns with their specific needs and concerns, increasing the likelihood of a successful therapeutic match.
* Flexible Communication Options: Betterhelp allows users to communicate with their therapists through text-based messaging, phone calls, or live video sessions. This flexibility allows users to choose the mode of communication that suits their comfort levels and communication preferences.
* Daily Messaging Support: Subscribers have the option to exchange unlimited text messages with their therapists throughout the week, providing continuous support and guidance. This frequent interaction can be particularly valuable for individuals who require ongoing assistance and real-time feedback.

Cons:

* Not Suitable for Crisis Situations: Betterhelp is designed for non-emergency situations. It may not be the best option for individuals experiencing severe mental health crises or those in need of immediate assistance. In such cases, in-person therapy or crisis helplines should be sought.
* Asynchronous Communication Limitations: The nature of text-based messaging may lead to slower-paced interactions compared to real-time conversations. Some users may find it challenging to express emotions effectively or experience difficulty in building rapport with their therapists.
* Limited Insurance Coverage: While Betterhelp offers subscription plans at various price points, some insurance providers may not cover the costs of online therapy services. This could make it less accessible and affordable for certain individuals.
* Potential Reliability Issues: As an online platform, Betterhelp relies on stable internet connections for effective communication. Technical issues or connectivity problems may disrupt sessions and impact the overall counseling experience.

Talkspace is an online therapy platform that offers mental health support and counseling services to individuals via text, voice, and video messaging. It aims to provide a convenient and accessible way for people to connect with licensed therapists and counselors from the comfort of their own homes. Founded in 2012, Talkspace has gained popularity as a prominent player in the teletherapy industry, catering to a wide range of mental health needs.

Pros:

* Convenience and Accessibility: Talkspace's platform enables users to access therapy from virtually anywhere with an internet connection. This convenience is particularly beneficial for individuals with limited mobility, busy schedules, or those living in remote areas without easy access to in-person counseling services.
* Anonymity and Privacy: Users have the option to remain anonymous or use a pseudonym during therapy sessions, providing a sense of privacy and confidentiality. This anonymity can be particularly appealing to those who feel hesitant or stigmatized about seeking therapy in traditional settings.
* Diverse Therapist Options: Talkspace offers a wide selection of licensed therapists and counselors with various specializations. This diversity allows users to find a mental health professional who best matches their specific needs and preferences, increasing the likelihood of building a strong therapeutic relationship.
* Flexible Communication: Talkspace offers multiple communication options, including text, voice, and video messaging. This flexibility allows users to choose the mode of communication that feels most comfortable for them, accommodating different communication styles and preferences.
* Continuous Access to Support: Users can communicate with their therapists on a daily basis, unlike traditional therapy sessions that typically occur on a weekly or bi-weekly basis. This frequent interaction can be particularly valuable during times of crisis or when ongoing support is needed.

Cons:

* Lack of Real-Time Interaction: While Talkspace provides timely responses from therapists, the nature of asynchronous messaging may lead to slower-paced conversations. Some individuals may prefer the immediacy of face-to-face interactions or live video sessions offered by traditional in-person therapy.
* Limited Non-Verbal Cues: Text-based communication can sometimes lack the non-verbal cues present in face-to-face interactions. This can make it challenging for therapists to fully understand and interpret the emotions and nuances conveyed by users.
* Not Suitable for Severe Conditions: Talkspace may not be the most suitable option for individuals with severe mental health conditions or those experiencing crisis situations. In-person therapy or more intensive treatment options might be necessary for such cases.
* Potential Miscommunication: The absence of real-time interaction can lead to miscommunications or misunderstandings between users and therapists, as tone and context may be difficult to convey through text-based messages.

## 2.2 The proposed system

The proposed system known as Icare, is a mobile application designed with an easy-to-use user interface which seeks to make navigation easy and needs little know how to utilize. This ensures that the users are able to relate more freely with the system and be able to use all its features to their utmost satisfaction.

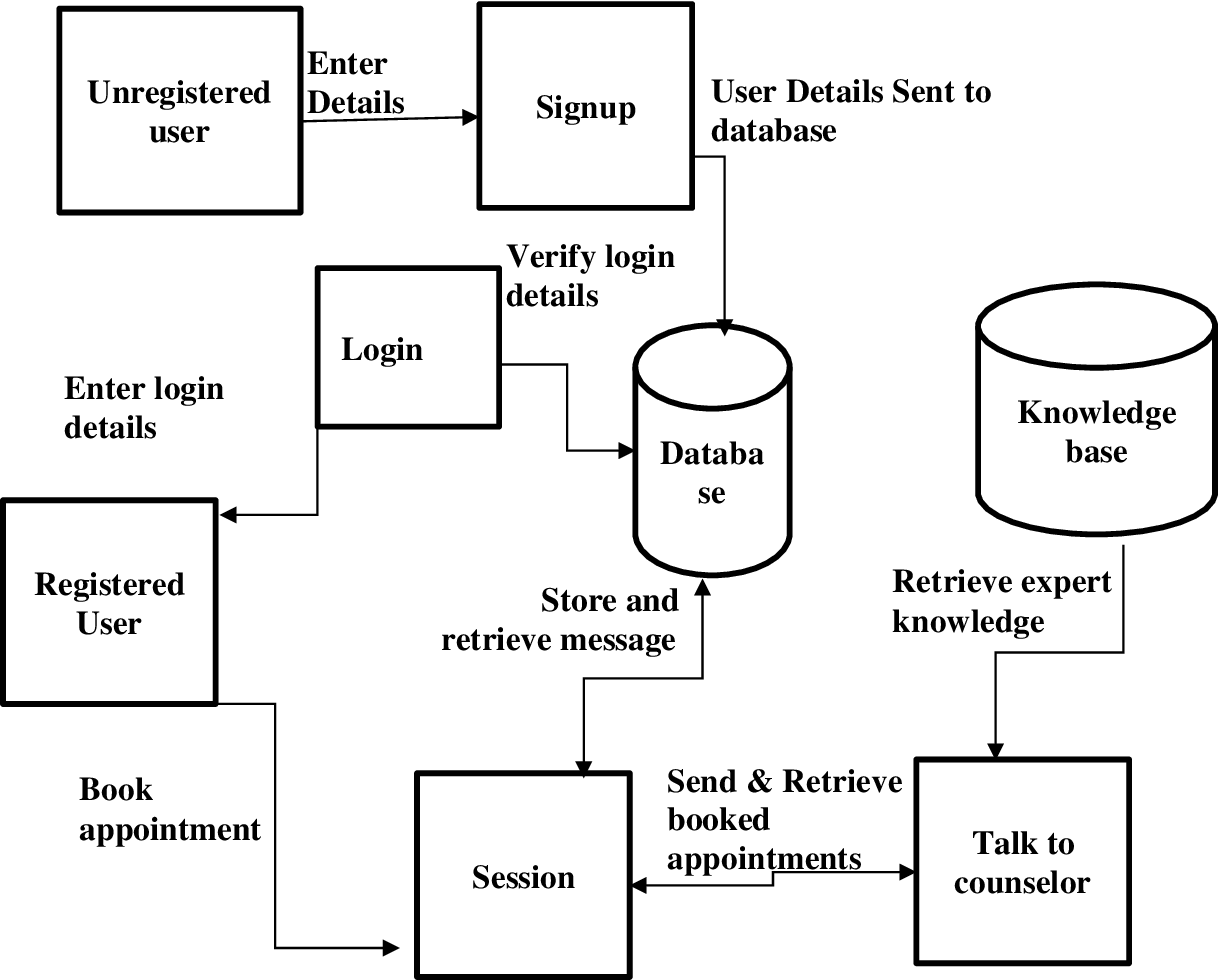
The key features are:

* Homepage: the home page screen coupled with both a menu and a tab bar for easy navigation. It also uses a slider to display various upcoming events and list views that display news concerning the counseling center. Providing real time information to users.
* Chat page: the chat page is equipped with an easy function to send messages to counselors through their emails. Giving specialized responses from counselors and not auto generated and repetitive replies as in the case of Replika.
* Call platform: The call platform helps you to conduct calls to counselors at the comfort of your home. This helps prevent asynchronous communication and gives a sense of emotion behind every reply given.
* Audio Platform: Which will hold approved audio messages for students to listen to at their own leisure to help them find ways to better their emotional health.
* Booking Platform: This where students can book appointment with counselors at the comfort of their hostels.
* User Authentication: This help keep their privacy secured and protected from unwanted intrusion.

These well thought of features, coupled with the ease of navigation will help extend the reach of the counseling center to different height and reach in helping student

## 2.3 Conceptual Design

Figure 2. 1. Conceptual of Design



## 2.4 Architecture of the proposed system.

Figure 2. 2. Architecture of proposed system

## 2.5 Components Designs and Components descriptions.

* Unregistered User: This refers to any fresher entering into KNUST, and trying to access the app. They will need to signup to have access into the provided system.
* Registered User: This refers to any continuing student in KNUST, accessing
* the app. They will need to They will to enter the required information to login into the provided system.
* The log in and signup: These components refer to the authentication screen which is displayed when user to try to enter the application.
* Sessions: refers to the times a counselor spends with the individual to help them with their emotional situation.
* Database: is the location where vital information are kept and hidden for verification and authentication.
* Knowledge base: Refers to the knowledge at the disposal of the counsellor.
* Counselor: Refers to a trained who sends feedback to students based on their messages.

## 2.6 Proposed system/software features.

The proposed mobile application will be equipped with the needed key features needed to make it an effective communication tool. These features mainly consist of:

* A user authentication interface which provides security for student information, ensures student confidentiality and creates trust between students and KNUST COUNSELLING CENTER (KCC). The user authentication will be handled by a backend database system will keep the required information need by students to log in each time the need to use the app. This information will include a student’s username or school e-mail address, index number and reference number provided by the school and only known by the student.
* Homepage with useful information embedded in the form of upcoming events and news. The upper portion will contain images in the form of card which has the aim of arousing student’s interest in the events being organized by the counseling center. The middle aspects will be made to display the latest news about the counselling center. Which will contain the latest achievements and accomplishments of the KNUST COUNSELLING CENNTER. This will student to openly trust the center as a place , where its guaranteed to locate and find help. The bottom aspect of the homepage will contain a tab bar which has all the key features including homepage itself to help with ease of navigation around the healthcare app.
* Chat platform which provides an avenue for students to be in close contact or communication level with their counsellors at the comfort of their hostels. This would help counsellors draw closer students who feel shy or uneasy about opening up and offer them the needed assistance. Each message sent by students will by received by the various counsellors in their respective colleges through their e-mail address. Giving both counsellors and any trained counselling personnel under the supervision of the counselling center the needed time, speed, accuracy and professional efficiency to attend to each individual.
* Call Floating button which gives student the chance of calling the counselling sector directly in emergency situations that require the school’s attention. The call page will be equipped with the active phone numbers of counsellors of the respective colleges across the KNUST campus and the call line of the KNUST security personnel to bring home the sense of safety around a student knowing that the school security services are a call away from assisting them.
* An audio platform with selected audio messages to aid student learn how to deal with situations in academics and audios from the student chaplaincy to help them in their spiritual growth. The growth of motivational speakers and influencers is rapidly rising on various social media platform across the internet. Though their goal is to assist people with their words, it’s not all that are beneficial. The audios on the app will be verified and provided by the counselling center and student’s chaplaincy.
* A booking platform which helps students book various appointments at their own comfort to interact with counsellors. The booking period would be within the working hours of the various counselling center at each college and working days. Each booked appointment will be sent to the emails of the counsellors, giving them a heads and prior notice of the students visit to their office to seek assistance.
* Backend database system that serves as a storage facility for keeping students log-in information such as password, school emails or username, reference number and index number and any vital information needed for utilizing the app to its full potential.

## 2.7 Development tools and environment

* The app will use flutter and dart as the main building base of the whole project. Therefore, the app UI will have and utilize to some extent the full potential and features provided by flutter and dart packages.
* It will also utilize a backend database system, provided by firebox which will keep the credentials of students needed to authenticate their identity before being allowed unto the app. This will help ensure safety and privacy of the user on the app.
* The app will be fully utilized on android devices as its main hardware components. Devices powered by android are massively used throughout KNUST campus and this will help with the wide spread and usage of the healthcare app to its fullest advantage by students. It will also make it available to students across campus and help the objectives and goals of KCC spread wide and far.
* The app utilizes visual studio code, which serves as it main IDE for implementing and bringing together all the needed packages from Flutter, Dart and Android Studio.
* Android Studio was used to provide a virtual android mobile device which served as the main texting environment on which the developed app was run.

## 2.8 Benefits of implementation of the proposed system.

* Provides an avenue where students seek advice concerning emotional stress and issues towards academic life and campus life.
* Gives the counseling center the avenue to interact with majority of students more intimately.
* Provides students the opportunity to book appointment corresponding to the free and available times or schedules of both the student and counselor respectively.
* The app provides a platform where the counseling center can reach out to majority of the student populace concerning any upcoming emotional and motivative program to help increase the knowledge of students concerning their emotional well-being.
* Ensures effective communication between the student and counselor in a short amount of time.

# CHAPTER 3

# **Methodology**

# 3.1Chapter Overview

This chapter of the documentation provides details required in the development and understanding of the project methodology. The project methodology, elaborates on the key elements such as requirement specification, stake holders of system, the requirement gathering process, functional requirements, UML diagrams, non-functional requirements, security concepts, project method, project design consideration and the developmental tools that goes a long way to describe the;

1. Details provided by the stake holders.
2. The various stake holders involved in the system.
3. The process used in gathering information concerning the requirement.
4. The specific functionalities and capabilities a system must possess to meet the needs and expectation of the stakeholders.
5. UML diagram to display the flow of logic within the project and Use Case Description to give details of the logic used.
6. System qualities and constraints that a system must possess to be considered successful in its overall performance.
7. Security properties of the project.
8. Project method and design considerations.
9. Developmental tools used in the creating the project.

## 3.2 Requirement specification

This defines detailed needs and expectations of stakeholders for a specific project. The various requirements provided by the stakeholders are:

* An app that provides an avenue for students to seek emotional and academic counseling.
* An app that provides a platform for the counseling center to engage students and make them aware of possible and upcoming events.
* An app that gives students the convenience of interacting with counselors at the comfort of the residence.
* An app that states and clearly defines the aims and goals of the counseling center to students.
* An app that makes it easier for both students and counselors to interact.
* An app to provide students with audio message to help them in their daily academic lives

These requirements provide a clear and concise description of what the project aims to achieve and what features and functionality it should possess to meet those objectives.

## 3.3 Stake holders of system

The stakeholders of the mobile application(project) are:

1. KNUST STUDENT POPULACE: The student populace of Kwame Nkrumah University of Science and Technology, play he primary role of stakeholder of the app by their daily interactions and the various activities they perform on the app.
2. KNUST COUNSELING CENTER: The various services required on the app by the student populace such as chatting a counselor, calling a counselor and booking an appointment will be provided by the counseling center to the students.
3. KNUST ADMINISTRATION: For an app which involves the student populace of KNUST and their counseling center to be implemented on the grounds or foundation of the school, it needs the full consideration, approval and acknowledgement from the school administrative body for it to be fully implemented.

## 3.4 Requirement Gathering process

The process in which various information concerning the acquisition of requirements needed from the mobile app were gathered through:

* Interaction with peer counselors: In other to have clear cut requirements and details concerning various requirements and their priorities, interaction with peer counselors was required to know what features to add to the app and what functionalities they should portray.
* Communication with the student populace: Communication with a handful of the student populace of KNUST was required to ascertain the requirements of students from the app and the counseling at large and how to solve them using the mobile app. And the various services they required from the app.
* Brainstorming and idea generation: Brainstorming and generating ideas as a fellow student helped me grasp some needed requirements and how to implement their solution in the app.
* Review and approval: Reviewing the various requirements by other of priority, combining similar requirements and approving of requirements helps give a clear goal as to the functions and requirement needed by the app for optimum functionality and approval.

## 3.5 Functional requirements

Functional requirements which define the needed specific functionalities and capabilities of the project to meet the needs and expectations of the stakeholders and describe what the system do and how it should behave under different conditions include:

1. User Authentication: The system shall provide a user authentication mechanism, allowing registered users to log in with a unique email or password.
2. Email: The email serves as a user authentication credential which gives a registered user asses into the system and its functionalities. The email also serves as a form of electronic transportation which delivers chat messages and appointment details to the email of the counselor.
3. Password: A unique password used as a credential in association with the email addresses provided to give users access to the system functionalities.
4. Data Entry: The system will allow users to enter data such as date, title, discussion, summary of appointment into the system to ensure flow of activity within the system.
5. Error Handling: The system shall display error messages when the user enters invalid inputs or responses into the system.
6. Notification System: The system will send messages and appointment messages to the appointed email designated to it within the app and inform students of possible feedbacks.

## 3.6 UML Diagrams

## 

Figure 3. 1. Use case diagram for the front-end models

## USE CASE DESCRIPTION

Use Case Name: ICARE MOBILE APPLICATION

Actors:

1.KNUST Counseling Center: The secondary actors that interact with students through the app and receives feedback from the system.

2.Students (Freshers and Continuing students): These are the primary actors who interact with the system and secondary actors through the system.

Use Cases:

Sign Up:

Description: This use case describes the process when a student creates a new account on the system.

Preconditions: None.

Postconditions: The student successfully creates an account and is logged into the system.

Main Flow:

The user opens the application.

The user clicks on the "Sign Up" button.

The system presents a registration form with fields for the student to enter their desired email, password and index number.

The student fills out the required information.

The student submits the registration form.

The system validates the information and checks for any existing accounts with the same email.

If the information is valid, the system creates a new account for the student.

The student receives a confirmation that the account has been successfully created.

The student is automatically logged into the system and directed to the home page.

Log In:

Description: This use case describes the process when a registered user logs into the mobile application.

Preconditions: The student must have a valid account on the platform.

Postconditions: The student is logged into the system and directed to the home page.

Main Flow:

The student opens the application.

The student clicks on the "Log In" button.

The system presents a login form with fields for the student to enter their email, password and index number.

The student enters their login credentials.

The student submits the login form.

The system verifies the provided credentials against the stored user data.

If the credentials are correct, the student is logged into the system.

The student is directed to the home page.

Home Page:

Description: This use case describes the main page of the application where the student can access various features.

Preconditions: The student must be logged into the system.

Postconditions: The student can access different pages and features within the platform.

Main Flow:

Upon successful login, the student is directed to the home page.

The home page displays an interface with options to access various features, such as chat, call, audio, and booking pages.

The student can click on the corresponding icons or buttons to navigate to the desired feature's page.

Chat Platform:

Description: This use case describes the page where information about the counseling center can be shared with the students.

Preconditions: The student must be logged into the system and be on the home page.

Postconditions: The student can interact with the page to view news and events.

Main Flow:

The student clicks on the "Chat" icon/button on the home page.

The system navigates the student to the chat page.

The chat page displays a list of existing chat conversations.

The system displays the selected chat conversation.

The student can send and receive messages in the chat platform.

Call Platform:

Description: This use case describes the page where the student can make voice calls to the counseling center.

Preconditions: The student must be logged into the system and be on the call platform.

Postconditions: The user can initiate and participate in voice calls.

Main Flow:

The user clicks on the "Call" icon/button on the home page.

The system navigates the user to the call page.

The call page displays a list of contacts.

The user can select a contact to make a call.

The system initiates the call and establishes the connection.

The user can engage in a voice call with the selected contact.

Audio Platform:

Description: This use case describes the page where the student can listen to audio approved by the counseling center.

Preconditions: The user must be logged into the system and be on the home page.

Postconditions: The user can listen to audio messages.

Main Flow:

The user clicks on the "Audio" icon/button on the home page.

The system navigates the user to the audio page.

The audio page displays a list of audio messages.

The user can select an audio message and listen.

Booking Page:

Description: This use case describes the page where the user can schedule appointments or bookings with the counseling center.

Preconditions: The user must be logged into the system and be on the home page.

Postconditions: The user can schedule and manage bookings.

Main Flow:

The user clicks on the "Booking" icon/button on the home page.

The system navigates the user to the booking page.

The booking page displays will display a form to fill and submit.

The user can select a date for the booking.

The user can send the booking request to the email of the counseling center.

Log Out:

Description: This use case describes the process when the user logs out of system.

Preconditions: The user must be logged into the system.

Postconditions: The user is logged out of the system and directed to the login page.

Main Flow:

The user clicks on the "Log Out" option with in the menu bar.

The system logs out the user and clears their active session.

The user is directed to the login page.

## 3.7 Non – functional requirements

This specify the quality attributes or systems qualities, specify the characteristic and constraints that a system must possess to be considered successful in its overall performance, rather than focusing on specific functionalities. Non-functional performances provided by the app and related to the overall user experience and the system’s ability to meet organizational and industry standards are:

* Performance: The app has a fast response time, speed and efficiency to meet the required delivery time of the user when the needed input and responses are made available.
* Usability: The app is built with an easy to understand and navigate user interface that allows for smooth user experience and utilization. This helps to ensure that fast and convenient advantage, even for new users.
* Scalability: The system is designed to handle the ever-increasing populace of the school with the firebase server system used as the primary online database server. This helps the app cope and still perform perfectly even with increasing number of students.
* Reliability: With the logical reasoning of the algorithm and error handling capabilities of the system. Users can rely on the system to provide the needed feedback.
* Compatibility: The app is compactible across android powered devices to ensure a wide range of reach to the student populace on campus.

## 3.8 Security concepts

* Confidentiality: The system ensures that sensitive information shared between both the counselor and Student are kept closed and only accessible by them. This involves controlling access to data and protecting it from disclosure to unauthorized users.
* Integrity: it ensures that data remains accurate and , complete and unaltered during transmission to the email of the counselor. It involves protecting data from unauthorized modifications, corruption or tampering.
* Availability: This ensures that the system resources and data are accessible to and usable when needed by authorized users.
* Authorization: This is the process of verifying the identity of users or devices attempting to access the system. It helps ensure that only legitimate users can gain access to resources based within the system.
* Multi-factor authentication: It enhances security by requiring users to provide multiple forms of identification before granting them access into the system. This typically includes the combination of an email address and a unique password available to the user.

## Project methods

The proposed system was built using the waterfall project methodology (plan-driven), which is a sequential approach in where each phase in developing the proposed application or system was well planned, executed, monitored and controlled to achieve the proposed objectives successfully before moving on to the next phase. The following stages were used in achieving the desired system:

* Requirements Gathering and Analysis: In a developing the system, there was a comprehensive upfront requirement gathering and analysis phase. Efforts were made to interact closely with stakeholders to gather all the necessary requirements for the application, including sign up, log in, home page, chat page, call page, audio page, booking page, and log out functionalities.
* Design: Once the requirements were gathered and analyzed, I moved on to the design phase. Detailed design documents for each page were created, defining how the user interface would look, how the different pages and features would be organized, and how the interactions between users and the system would be implemented.
* Implementation: After completing the design phase, the actual coding and implementation of the application began. Each interface, such as sign up, log in, chat page, call page, etc., would be implemented based on the design specifications.
* Testing: The developed functionalities were thoroughly tested to ensure they meet the specified requirements. This includes unit testing, integration testing, system testing, and user acceptance testing.
* Deployment: Once all the testing is successfully completed, the application was deployed to on a virtual phone.
* Maintenance: After deployment, any subsequent changes, updates, or bug fixes would follow a predefined change management process.

## Chosen model and justification.

The waterfall project method was used due to its systematic approach and thorough phase analysis. The system needed to have well defined and analyzed requirements in order to satisfy the required needs. The waterfall project methodology, gives specific strategies in the development of systems such as icare, which need laid out methods to follow in other to complete a previous method and move on to the next one. It also gives ways of testing the project with predefined parameter and expected out comes needed from the developed system.

## PROJECT DESIGN CONSIDERATION (LOGICAL DESIGNS)

## UI Design

* User Authentication UI: The user authentication user interface comes with a green background, the name of the application and three text fields that take in the user’s email, password, and index number respectively. It also has a log in and sign up button located at the bottom of the text field form interface which has a white background.
* Homepage: The homepage has a green app bar which has a menu on its top left corner which comes with key features such as the home, chat, call, audio, book and log out for easy navigation throughout the application. It also possess a green tab bar at the bottom with similar key features except logout. The home page contains a white background on which a sliding event card displays all the upcoming events planned and a section for displaying news related to the counseling center.
* Chat Platform: The chat platform has the same color features as the home screen i.e., green app and white background. It come with a text located at the bottom center of the page . this text button brings out a form to chat with the counselor. These messages are then displayed as list tiles on the screen including replies from the counselor.
* Call Platform: The call platform comes with the symbol of both the counseling center and the school located at the top of the page with and the numbers of the counseling center located below it and separated by a divider. These numbers are displayed as list tiles with a call icon as the leading, the number as the title and a call button as the end in the tile.
* Audio Platform: the audio page shows a list of audio message arranged in grid form, from which the user could select and play any audio of their liking.
* Booking Platform: The booking platform is similar to the chat platform with the text button located at the bottom center, which brings up a form to book an appointment with the counselor. Both the chat and booking platform relay the provided messages through email to the counselors.

## DB Design

## 

## 

Figure 3. 2. E-R DIAGRAMS

STUDENTS

UserId

Email

Index Number

Password

COUNSELOR

Password

Email

Figure 3. 3 DB SCHEMA

## Developmental tools

* The app will use flutter and dart as the main building base of the whole project. Therefore, the app UI will have and utilize to some extent the full potential and features provided by flutter and dart packages.
* It will also utilize a backend database system, provided by firebox which will keep the credentials of students needed to authenticate their identity before being allowed unto the app. This will help ensure safety and privacy of the user on the app.
* The app will be fully utilized on android devices as its main hardware components. Devices powered by android are massively used throughout KNUST campus and this will help with the wide spread and usage of the healthcare app to its fullest advantage by students. It will also make it available to students across campus and help the objectives and goals of KCC spread wide and far.
* The app utilizes visual studio code, which serves as it main IDE for implementing and bringing together all the needed packages from Flutter, Dart and Android Studio.
* Android Studio was used to provide a virtual android mobile device which served as the main texting environment on which the developed app was run.

# CHAPTER 4

# IMPLEMENTATION AND RESULTS

## Chapter Overview.

This chapter elaborate on the various implementations used in designing the project and results gotten from various text and analysis. The aim of this chapter is to outline the implementations, results of applying them, testing of the application, the testing plan which is categorized into component testing and system test and mapping of the logical design onto physical platforms.

The project implementation phase included key steps to help in arching the required and desired results. These steps typically included:

* Task Assignment: Specific tasks had to be assigned in other to achieve the required results complete on deadline. Task such as requirement gathering, scheduling time to implement codes and analyze growth of project and research to gather the latest information to use in my project to help acquire the needed and desired results.
* Resource Allocation: The allocation of resources such as time, money and human resources etc. had to be put in play to achieve results. The time resource was the number of hours used in writing code, implementing features, analyzing achieved project with requirements in hand. The human resource was the effort made to gather the needed information using various interview methods and money in terms of data bundle for various upgrades in system, research and running numerous text.
* Monitoring and Control: Continuously monitoring the rapid growth and progress of the system and quickly addressing any issues promptly and accurately.
* Quality Control and Assurance: Having predefined aims, objectives and requirements serve as a guide in the development of the project having in mind the required and final outcome of the project.
* Testing and Validation: Conducting thorough testing and validation of the project deliverables to ensure they meet the requirement and expectation.

## Mapping logical design onto physical platform

## algorithm for implementing UI

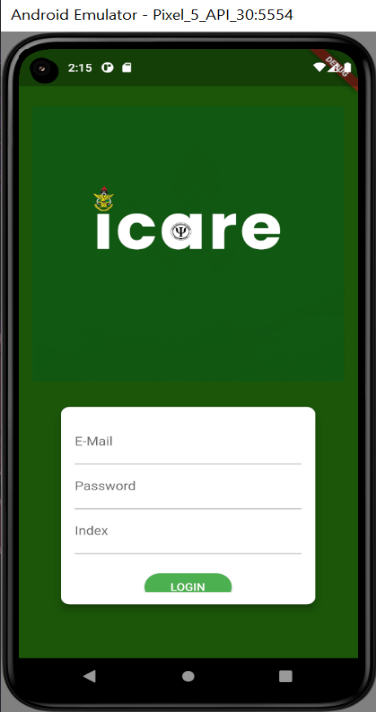
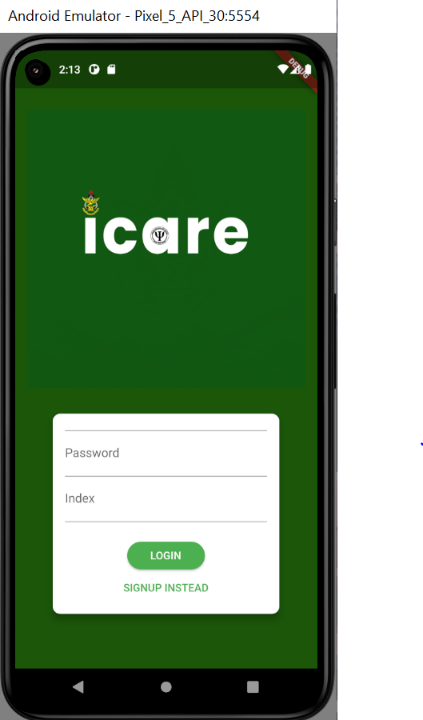


Figure 4. 1. **Login and Signup Screen**

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

import '../provider/auth.dart';

import '../models/http\_exceptions.dart';

enum AuthMode { Signup, Login }

class AuthScreen extends StatelessWidget {

  const AuthScreen({super.key});

  static const routeName = '/auth';

  @override

  Widget build(BuildContext context) {

    final deviceSize = MediaQuery.of(context).size;

    return Scaffold(

      backgroundColor: Color.fromARGB(255, 27, 86, 9),

      body: Stack(children: [

        SingleChildScrollView(

          child: Container(

            height: deviceSize.height,

            width: deviceSize.width,

            child: Column(

              mainAxisAlignment: MainAxisAlignment.center,

              crossAxisAlignment: CrossAxisAlignment.center,

              children: [

                Flexible(

                  child: Container(

                    margin: EdgeInsets.only(bottom: 20),

                    padding: EdgeInsets.all(10),

                    child: Image.asset('assets/images/icare2.jpg'),

                  ),

                ),

                Flexible(

                    flex: deviceSize.width > 600 ? 2 : 1, child: AuthCard())

              ],

            ),

          ),

        )

      ]),

    );

  }

}

class AuthCard extends StatefulWidget {

  const AuthCard({super.key});

  @override

  State<AuthCard> createState() => \_AuthCardState();

}

class \_AuthCardState extends State<AuthCard> {

  final GlobalKey<FormState> \_formKey = GlobalKey();

  AuthMode \_authMode = AuthMode.Login;

  Map<String, String> \_authData = {

    'email': '',

    'password': '',

    'indexNumber': ''

  };

  var \_isLoading = false;

  final \_passwordController = TextEditingController();

  final \_indexNumberController = TextEditingController();

  void \_showErrorDialog(String message) {

    showDialog(

        context: context,

        builder: (ctx) => AlertDialog(

              title: Text('An error occured'),

              content: Text(message),

              actions: [

                TextButton(

                  onPressed: Navigator.of(context).pop,

                  child: Text(

                    'Okay',

                  ),

                )

              ],

            ));

  }

  Future<void> \_submit() async {

    if (!\_formKey.currentState!.validate()) {

      //Invalid

      return;

    }

    \_formKey.currentState!.save();

    setState(() {

      \_isLoading = true;

    });

    try {

      if (\_authMode == AuthMode.Login) {

        //Log user in

        await Provider.of<Auth>(context, listen: false).login(

            \_authData['email'].toString(),

            \_authData['password'].toString(),

            \_authData['indexNumber'].toString());

      } else {

        //Sign user up

        await Provider.of<Auth>(context, listen: false).signup(

            \_authData['email'].toString(),

            \_authData['password'].toString(),

            \_authData['indexNumber'].toString());

      }

    } on HttpException catch (error) {

      var errorMessage = 'Authentication failed';

      if (error.toString().contains('EMAIL\_EXISTS')) {

        errorMessage = 'This email address is already in use.';

      } else if (error.toString().contains('INVALID\_EMAIL')) {

        errorMessage = 'This is not valid email address';

      } else if (error.toString().contains('WEAK\_PASSWORD')) {

        errorMessage = 'This password is weak';

      } else if (error.toString().contains('EMAIL\_NOT\_FOUND')) {

        errorMessage = 'Could not find a user with that email.';

      } else if (error.toString().contains('INVALID\_PASSWORD')) {

        errorMessage = 'Invalid password';

        print(error);

      }

      \_showErrorDialog(errorMessage);

    } catch (error) {

      const errorMessage = 'Could not authenticate you. Please try again later';

      \_showErrorDialog(errorMessage);

    }

    setState(() {

      \_isLoading = false;

    });

  }

  void \_switchAuthMode() {

    if (\_authMode == AuthMode.Login) {

      setState(() {

        \_authMode = AuthMode.Signup;

      });

    } else {

      setState(() {

        \_authMode = AuthMode.Login;

      });

    }

  }

  @override

  Widget build(BuildContext context) {

    final deviceSize = MediaQuery.of(context).size;

    return Card(

      shape: RoundedRectangleBorder(

        borderRadius: BorderRadius.circular(10.0),

      ),

      elevation: 8.0,

      child: Container(

        height: \_authMode == AuthMode.Signup ? 320 : 260,

        constraints: BoxConstraints(

          minHeight: \_authMode == AuthMode.Signup ? 320 : 260,

        ),

        width: deviceSize.width \* 0.75,

        padding: EdgeInsets.all(16.0),

        child: Form(

            key: \_formKey,

            child: SingleChildScrollView(

              child: Column(

                children: [

                  TextFormField(

                      decoration: InputDecoration(

                        labelText: 'E-Mail',

                      ),

                      keyboardType: TextInputType.emailAddress,

                      validator: (value) {

                        if (value!.isEmpty || !value.contains('@')) {

                          return 'Invalid email';

                        }

                      },

                      onSaved: (value) {

                        \_authData['email'] = value.toString();

                      }),

                  TextFormField(

                    decoration: InputDecoration(labelText: 'Password'),

                    obscureText: true,

                    controller: \_passwordController,

                    validator: (value) {

                      if (value!.isEmpty || value.length < 8) {

                        return 'Pasword is too short';

                      }

                    },

                    onSaved: (newValue) =>

                        \_authData['password'] = newValue.toString(),

                  ),

                  if (\_authMode == AuthMode.Signup)

                    TextFormField(

                      decoration:

                          InputDecoration(labelText: 'Confirm Password'),

                      obscureText: true,

                      validator: \_authMode == AuthMode.Signup

                          ? (value) {

                              if (value != \_passwordController.text) {

                                return 'Pasword do not match';

                              }

                            }

                          : null,

                      onSaved: (newValue) =>

                          \_authData['password'] = newValue.toString(),

                    ),

                  TextFormField(

                    decoration: InputDecoration(labelText: 'Index'),

                    obscureText: true,

                    controller: \_indexNumberController,

                    validator: (value) {

                      if (value!.isEmpty || value.length < 8) {

                        return;

                      }

                    },

                    onSaved: (newValue) =>

                        \_authData['indexNumber'] = newValue.toString(),

                  ),

                  SizedBox(

                    height: 20,

                  ),

                  if (\_isLoading)

                    CircularProgressIndicator()

                  else

                    ElevatedButton(

                      onPressed: \_submit,

                      child: Text(

                          \_authMode == AuthMode.Login ? 'LOGIN' : 'SIGN UP'),

                      style: ElevatedButton.styleFrom(

                        shape: RoundedRectangleBorder(

                            borderRadius: BorderRadius.circular(20)),

                        padding: EdgeInsets.symmetric(

                            horizontal: 30.0, vertical: 8.0),

                        // remember to color buttom

                      ),

                    ),

                  TextButton(

                      onPressed: \_switchAuthMode,

                      child: Text(

                        '${\_authMode == AuthMode.Login ? 'SIGNUP' : 'LOGIN'} INSTEAD',

                      ),

                      style: TextButton.styleFrom(

                          padding: EdgeInsets.symmetric(

                              horizontal: 30.0, vertical: 4.0),

                          tapTargetSize: MaterialTapTargetSize.shrinkWrap

                          //do color

                          ))

                ],

              ),

            )),

      ),

    );

  }

}

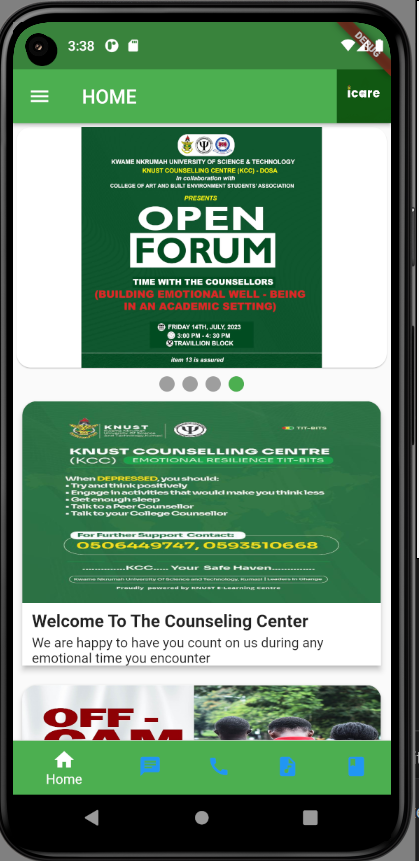


Figure 4. 2. Home Screen

import 'package:flutter/material.dart';

import '../widgets/event\_slider.dart';

import '../models/news.dart';

class HomepageScreen extends StatelessWidget {

  static const routeName = './home';

  final List<News> \_news = [

    News(

      id: 'c1',

      title: 'Welcome To The Counseling Center',

      subject:

          'We are happy to have you count on us during any emotional time you encounter',

      imageUrl: 'assets/images/flyer.jpg',

    ),

    News(

      id: 'c2',

      title: 'Off Campus Visitation',

      subject:

          'We bring the counseling center to your door step and at your comfort',

      imageUrl: 'assets/event\_flyers/flyer\_2.jpg',

    )

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Column(

        children: [

          //Events session

          EventSlider(),

          Container(

            height: 362,

            child: ListView.builder(

              itemBuilder: ((context, index) => Card(

                    shape: const RoundedRectangleBorder(

                      borderRadius: BorderRadius.only(

                        topLeft: Radius.circular(15),

                        topRight: Radius.circular(15),

                      ),

                    ),

                    elevation: 5,

                    margin: const EdgeInsets.all(10),

                    child: Column(

                      crossAxisAlignment: CrossAxisAlignment.stretch,

                      children: [

                        AspectRatio(

                          aspectRatio: 16 / 9,

                          child: ClipRRect(

                            borderRadius: const BorderRadius.only(

                              topLeft: Radius.circular(15),

                              topRight: Radius.circular(15),

                            ),

                            child: Image.asset(

                              \_news[index].imageUrl,

                              fit: BoxFit.fill,

                            ),

                          ),

                        ),

                        const SizedBox(height: 8),

                        Padding(

                          padding: const EdgeInsets.symmetric(horizontal: 10),

                          child: Column(

                            crossAxisAlignment: CrossAxisAlignment.start,

                            children: [

                              Text(

                                \_news[index].title,

                                style: const TextStyle(

                                  fontSize: 18,

                                  fontWeight: FontWeight.bold,

                                ),

                              ),

                              const SizedBox(height: 4),

                              Text(

                                \_news[index].subject,

                                style: const TextStyle(fontSize: 14),

                              ),

                            ],

                          ),

                        ),

                      ],

                    ),

                  )),

              itemCount: \_news.length,

            ),

          )

        ],

      ),

    );

  }

}

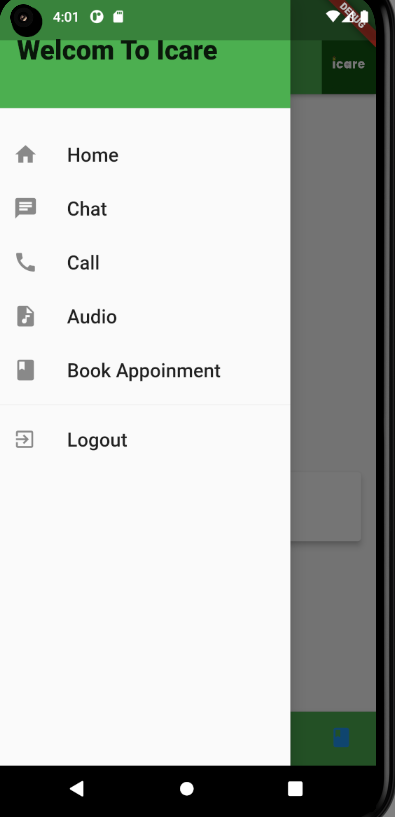


Figure 4. 3. Menu Bar

import 'package:flutter/material.dart';

import 'package:mini\_project/screens/audio\_screen.dart';

import 'package:mini\_project/screens/booking\_screen.dart';

import 'package:mini\_project/screens/call\_screen.dart';

import 'package:mini\_project/screens/chart\_screen.dart';

import 'package:mini\_project/screens/tabs\_screen.dart';

import 'package:provider/provider.dart';

import '../provider/auth.dart';

class MAINDRAWERS extends StatelessWidget {

  const MAINDRAWERS({super.key});

  Widget buildListTile(title, IconData icon, Function tapHandler) {

    return ListTile(

      leading: Icon(

        icon,

        size: 26,

      ),

      title: Text(

        title,

        style: TextStyle(fontSize: 20),

      ),

      onTap: (() => tapHandler),

    );

  }

  @override

  Widget build(BuildContext context) {

    return Drawer(

      child: Column(children: [

        Container(

          height: 120,

          width: double.infinity,

          padding: const EdgeInsets.all(20),

          alignment: Alignment.centerLeft,

          color: Theme.of(context).primaryColor,

          child: Text(

            'Welcom To Icare',

            style: TextStyle(

              fontWeight: FontWeight.w900,

              fontSize: 28,

            ),

          ),

        ),

        const SizedBox(

          height: 20,

        ),

        buildListTile('Home', Icons.home, () {

          Navigator.of(context).pushNamed(TabsScreen.routeName);

        }),

        buildListTile('Chat', Icons.chat,

            () => Navigator.of(context).pushNamed(ChatsScreen.routeName)),

        buildListTile('Call', Icons.call, () {

          Navigator.of(context).pushNamed(CallScreen.routeName);

        }),

        buildListTile('Audio', Icons.audio\_file, () {

          Navigator.of(context).pushNamed(AudioScreen.routeName);

        }),

        buildListTile('Book Appoinment', Icons.book, () {

          Navigator.of(context).pushNamed(BookingScreen.routeName);

        }),

        Divider(),

        ListTile(

          leading: Icon(Icons.exit\_to\_app),

          title: Text(

            'Logout',

            style: TextStyle(fontSize: 20),

          ),

          onTap: () {

            Navigator.of(context).pop();

            Provider.of<Auth>(context, listen: false).logout();

          },

        )

      ]),

    );

  }

}

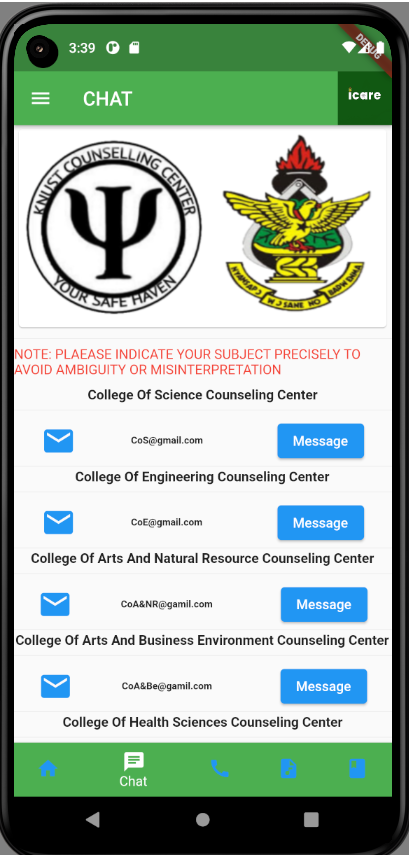


Figure 4. 4. Chat Screen

import 'package:flutter/material.dart';

import 'package:url\_launcher/url\_launcher.dart';

class ChatsScreen extends StatelessWidget {

  static const routeName = '/chats\_screen';

  final List<Map<String, String>> emailReceipient = [

    {'CoS': 'CoS@gmail.com.com'},

    {'CoE': 'CoE@gmail.com'},

    {'COA&NR': 'COA&NS@gamil.com'},

    {'COA&Be': 'COA&Be@gamil.com'},

    {'COHS': 'COHS@gamil.com'},

    {'COH&SS': 'COH&SS@gamil.com'}

  ];

  ChatsScreen({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: SingleChildScrollView(

        child: Column(

          children: [

            Card(

              child: ClipRRect(

                  borderRadius: const BorderRadius.all(Radius.circular(20)),

                  child: Image.asset('assets/images/kcc-knust.png')),

            ),

            const Divider(),

            const Text(

              'NOTE: PLAEASE INDICATE YOUR SUBJECT PRECISELY TO AVOID AMBIGUITY OR MISINTERPRETATION',

              style: TextStyle(color: Colors.red),

            ),

            const SizedBox(

              height: 10,

            ),

            Column(children: [

              const Text(

                'College Of Science Counseling Center',

                style: TextStyle(fontWeight: FontWeight.bold),

              ),

              const Divider(),

              Row(

                mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                children: [

                  const Icon(

                    Icons.email,

                    size: 36,

                    color: Colors.blue,

                  ),

                  const SizedBox(

                    height: 20,

                  ),

                  const Text(

                    'CoS@gmail.com',

                    style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                    softWrap: true,

                  ),

                  const SizedBox(

                    width: 20,

                  ),

                  ElevatedButton(

                    onPressed: () async {

                      String? encodeQueryParameters(

                          Map<String, String> params) {

                        return params.entries

                            .map((MapEntry<String, String> e) =>

                                '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                            .join('&');

                      }

                      final Uri emailLaunchUri = Uri(

                        scheme: 'mailto',

                        path: 'princekoomson03@gmail.com',

                        query: encodeQueryParameters(<String, String>{

                          'subject': 'Hello',

                          'body': 'How may we help you'

                        }),

                      );

                      if (await canLaunchUrl(emailLaunchUri)) {

                        launchUrl(emailLaunchUri);

                      } else {

                        throw Exception('Could not launch $emailLaunchUri');

                      }

                      try {

                        (await canLaunchUrl(emailLaunchUri));

                      } catch (e) {

                        print(e.toString);

                      }

                    },

                    style:

                        ElevatedButton.styleFrom(backgroundColor: Colors.blue),

                    child: const Text('Message'),

                  ),

                ],

              ),

            ]),

            //end

            const Divider(

              height: 5,

            ),

            Column(

              children: [

                const Text(

                  'College Of Engineering Counseling Center',

                  style: TextStyle(fontWeight: FontWeight.bold),

                ),

                const Divider(),

                Row(

                  mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                  children: [

                    const Icon(

                      Icons.email,

                      size: 36,

                      color: Colors.blue,

                    ),

                    const SizedBox(

                      height: 20,

                    ),

                    const Text(

                      'CoE@gmail.com',

                      style:

                          TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                      softWrap: true,

                    ),

                    const SizedBox(

                      width: 20,

                    ),

                    ElevatedButton(

                      onPressed: () async {

                        String? encodeQueryParameters(

                            Map<String, String> params) {

                          return params.entries

                              .map((MapEntry<String, String> e) =>

                                  '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                              .join('&');

                        }

                        final Uri emailLaunchUri = Uri(

                          scheme: 'mailto',

                          path: 'princekoomson03@gmail.com',

                          query: encodeQueryParameters(<String, String>{

                            'subject': '',

                          }),

                        );

                        if (await canLaunchUrl(emailLaunchUri)) {

                        } else {

                          throw Exception('Could not launch $emailLaunchUri');

                        }

                        launchUrl(emailLaunchUri);

                      },

                      style: ElevatedButton.styleFrom(

                          backgroundColor: Colors.blue),

                      child: const Text('Message'),

                    ),

                  ],

                ),

              ],

            ),

            //end,

            const Divider(

              height: 5,

            ),

            Column(

              mainAxisAlignment: MainAxisAlignment.spaceEvenly,

              children: [

                const Text(

                  'College Of Arts And Natural Resource Counseling Center',

                  style: TextStyle(fontWeight: FontWeight.bold),

                ),

                const Divider(),

                Row(

                  mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                  children: [

                    const Icon(

                      Icons.email,

                      size: 36,

                      color: Colors.blue,

                    ),

                    const SizedBox(

                      height: 20,

                    ),

                    const Text(

                      'CoA&NR@gamil.com',

                      style:

                          TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                    ),

                    const SizedBox(

                      width: 20,

                    ),

                    ElevatedButton(

                      onPressed: () async {

                        String? encodeQueryParameters(

                            Map<String, String> params) {

                          return params.entries

                              .map((MapEntry<String, String> e) =>

                                  '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                              .join('&');

                        }

                        final Uri emailLaunchUri = Uri(

                          scheme: 'mailto',

                          path: 'princekoomson03@gmail.com',

                          query: encodeQueryParameters(<String, String>{

                            'subject': '',

                          }),

                        );

                        if (await canLaunchUrl(emailLaunchUri)) {

                        } else {

                          throw Exception('Could not launch $emailLaunchUri');

                        }

                        launchUrl(emailLaunchUri);

                      },

                      style: ElevatedButton.styleFrom(

                          backgroundColor: Colors.blue),

                      child: const Text('Message'),

                    ),

                  ],

                ),

              ],

            ),

            //end,

            const Divider(

              height: 5,

            ),

            Column(

              children: [

                const Text(

                  'College Of Arts And Business Environment Counseling Center',

                  style: TextStyle(fontWeight: FontWeight.bold),

                ),

                const Divider(),

                Row(

                  mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                  children: [

                    const Icon(

                      Icons.email,

                      size: 36,

                      color: Colors.blue,

                    ),

                    const SizedBox(

                      height: 20,

                    ),

                    const Text(

                      'CoA&Be@gamil.com',

                      style:

                          TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                    ),

                    const SizedBox(

                      width: 20,

                    ),

                    ElevatedButton(

                      onPressed: () async {

                        String? encodeQueryParameters(

                            Map<String, String> params) {

                          return params.entries

                              .map((MapEntry<String, String> e) =>

                                  '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                              .join('&');

                        }

                        final Uri emailLaunchUri = Uri(

                          scheme: 'mailto',

                          path: 'princekoomson03@gmail.com',

                          query: encodeQueryParameters(<String, String>{

                            'subject': '',

                          }),

                        );

                        if (await canLaunchUrl(emailLaunchUri)) {

                        } else {

                          throw Exception('Could not launch $emailLaunchUri');

                        }

                        launchUrl(emailLaunchUri);

                      },

                      style: ElevatedButton.styleFrom(

                          backgroundColor: Colors.blue),

                      child: const Text('Message'),

                    ),

                  ],

                ),

              ],

            ),

            //end,

            const Divider(

              height: 5,

            ),

            Column(

              children: [

                const Text(

                  'College Of Health Sciences Counseling Center',

                  style: TextStyle(fontWeight: FontWeight.bold),

                ),

                const Divider(),

                Row(

                  mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                  children: [

                    const Icon(

                      Icons.email,

                      size: 36,

                      color: Colors.blue,

                    ),

                    const SizedBox(

                      height: 20,

                    ),

                    const Text(

                      'COHS@gamil.com',

                      style:

                          TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                    ),

                    const SizedBox(

                      width: 20,

                    ),

                    ElevatedButton(

                      onPressed: () async {

                        String? encodeQueryParameters(

                            Map<String, String> params) {

                          return params.entries

                              .map((MapEntry<String, String> e) =>

                                  '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                              .join('&');

                        }

                        final Uri emailLaunchUri = Uri(

                          scheme: 'mailto',

                          path: 'princekoomson03@gmail.com',

                          query: encodeQueryParameters(<String, String>{

                            'subject': '',

                          }),

                        );

                        if (await canLaunchUrl(emailLaunchUri)) {

                        } else {

                          throw Exception('Could not launch $emailLaunchUri');

                        }

                        launchUrl(emailLaunchUri);

                      },

                      style: ElevatedButton.styleFrom(

                          backgroundColor: Colors.blue),

                      child: const Text('Message'),

                    ),

                  ],

                ),

              ],

            ),

            //end,

            const Divider(

              height: 5,

            ),

            Column(

              children: [

                const Text(

                  'College Of Health And Social Sciences Counseling Center',

                  style: TextStyle(fontWeight: FontWeight.bold),

                ),

                const Divider(),

                Row(

                  mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                  children: [

                    const Icon(

                      Icons.email,

                      size: 36,

                      color: Colors.blue,

                    ),

                    const SizedBox(

                      height: 20,

                    ),

                    const Text(

                      'COH&SS@gamil.com',

                      style:

                          TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                    ),

                    const SizedBox(

                      width: 20,

                    ),

                    ElevatedButton(

                      onPressed: () async {

                        String? encodeQueryParameters(

                            Map<String, String> params) {

                          return params.entries

                              .map((MapEntry<String, String> e) =>

                                  '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                              .join('&');

                        }

                        final Uri emailLaunchUri = Uri(

                          scheme: 'mailto',

                          path: 'princekoomson03@gmail.com',

                          query: encodeQueryParameters(<String, String>{

                            'subject': '',

                          }),

                        );

                        if (await canLaunchUrl(emailLaunchUri)) {

                        } else {

                          throw Exception('Could not launch $emailLaunchUri');

                        }

                        launchUrl(emailLaunchUri);

                      },

                      style: ElevatedButton.styleFrom(

                          backgroundColor: Colors.blue),

                      child: const Text('Message'),

                    ),

                  ],

                ),

              ],

            )

            //end,

          ],

        ),

      ),

    );

  }

}

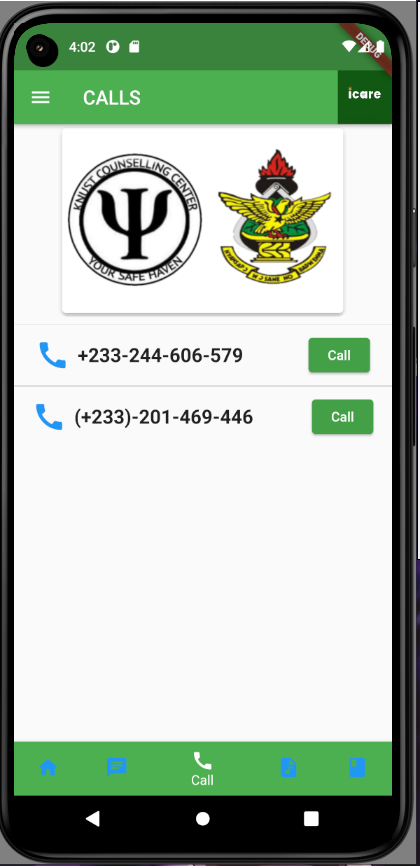
import 'package:flutter/material.dart';

Figure 4. 5 Call Platform

import 'package:url\_launcher/url\_launcher.dart';

class CallScreen extends StatelessWidget {

  const CallScreen({super.key});

  static const routeName = '/ call\_screen';

  Widget buildCall(IconData icon, String number, String prefix) {

    return Row(

      mainAxisAlignment: MainAxisAlignment.spaceEvenly,

      children: [

        Row(

          children: [

            Icon(

              icon,

              size: 36,

              color: Colors.blue,

            ),

            const SizedBox(

              width: 8,

            ),

            SelectableText(

              number,

              style: const TextStyle(

                fontSize: 20,

                fontWeight: FontWeight.bold,

              ),

            ),

          ],

        ),

        SizedBox(

          width: 20,

        ),

        ElevatedButton(

            onPressed: () async {

              final Uri url = Uri(

                scheme: prefix,

                path: number,

              );

              if (await canLaunchUrl(url)) {

                await launchUrl(url);

              } else {

                print('cannot Launch this url');

              }

            },

            style: ElevatedButton.styleFrom(

              backgroundColor: Colors.green.shade600,

            ),

            child: const Text('Call'))

      ],

    );

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Column(

        children: [

          ClipRRect(

            borderRadius: BorderRadius.all(Radius.circular(10)),

            child: Container(

              width: 300,

              height: 200,

              child: Card(

                elevation: 5,

                child: Image.asset('assets/images/kcc-knust.png'),

              ),

            ),

          ),

          Divider(),

          // Call logs

          buildCall(Icons.call, '+233-244-606-579', 'tel'),

          const Divider(

            thickness: 2,

          ),

          buildCall(Icons.call, '(+233)-201-469-446', 'tel')

        ],

      ),

    );

  }

}

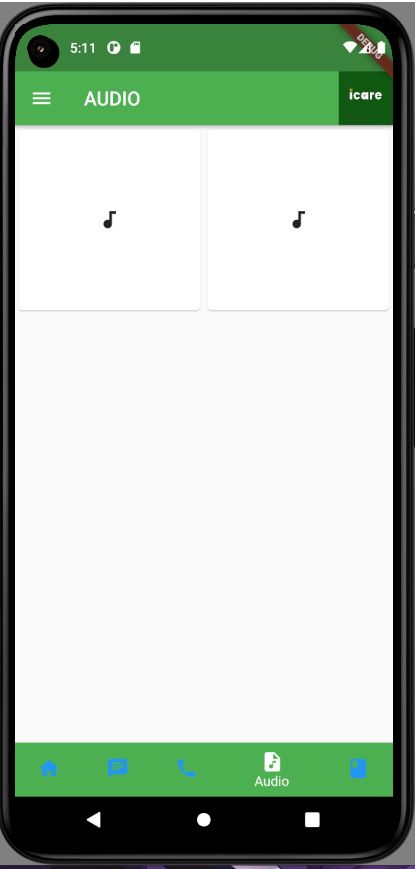


Figure 4. 6. Audio Platform

import 'package:flutter/material.dart';

import 'package:audioplayers/audioplayers.dart';

class AudioScreen extends StatelessWidget {

  static const routeName = '/audio\_screen';

  final List<String> audioFiles = [

    'audio/audio\_one.mp3',

    'audio/mighty\_one.mp3',

    // Add more audio file paths here

  ];

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: GridView.builder(

        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(

          crossAxisCount: 2, // Number of columns in the grid

        ),

        itemCount: audioFiles.length,

        itemBuilder: (context, index) {

          return GestureDetector(

            onTap: () => \_playAudio(context, audioFiles[index]),

            child: Card(

              child: Center(

                child: Icon(Icons.audiotrack),

              ),

            ),

          );

        },

      ),

    );

  }

  void \_playAudio(BuildContext context, String audioFilePath) {

    final player = AudioPlayer();

    player.play(AssetSource(audioFilePath));

  }

}

## 

## 

Figure 4. 7 Booking Platform

import 'package:flutter/material.dart';

import 'package:url\_launcher/url\_launcher.dart';

class BookingScreen extends StatelessWidget {

  static const routeName = '/chats\_screen';

  final List<Map<String, String>> emailReceipient = [

    {'CoS': 'CoS@gmail.com'},

    {'CoE': 'CoE@gamil.com'},

    {'COA&NR': 'COA&NS@gamil.com'},

    {'COA&Be': 'COA&Be@gamil.com'},

    {'COHS': 'COHS@gamil.com'},

    {'COH&SS': 'COH&SS@gamil.com'}

  ];

  BookingScreen({super.key});

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: SingleChildScrollView(

        child: Column(children: [

          Card(

            child: ClipRRect(

                borderRadius: const BorderRadius.all(Radius.circular(20)),

                child: Image.asset('assets/images/kcc-knust.png')),

          ),

          const SizedBox(

            height: 10,

          ),

          Column(children: [

            const Text(

              'College Of Science Counseling Center',

              style: TextStyle(fontWeight: FontWeight.bold),

            ),

            const Divider(),

            Row(

              mainAxisAlignment: MainAxisAlignment.spaceEvenly,

              children: [

                const Icon(

                  Icons.email,

                  size: 36,

                  color: Colors.blue,

                ),

                const SizedBox(

                  height: 20,

                ),

                const Text(

                  'CoS@gmail.com',

                  style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                ),

                const SizedBox(

                  width: 20,

                ),

                ElevatedButton(

                  onPressed: () {

                    String? encodeQueryParameters(Map<String, String> params) {

                      return params.entries

                          .map((MapEntry<String, String> e) =>

                              '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                          .join('&');

                    }

                    final Uri emailLaunchUri = Uri(

                      scheme: 'mailto',

                      path: emailReceipient[0].toString(),

                      query: encodeQueryParameters(<String, String>{

                        'subject': 'Booking A COUNSELING Appointment',

                      }),

                    );

                    launchUrl(emailLaunchUri);

                  },

                  style: ElevatedButton.styleFrom(backgroundColor: Colors.blue),

                  child: const Text('Book'),

                ),

              ],

            ),

          ]),

          const Divider(

            height: 30,

            thickness: 5,

          ),

          Column(

            children: [

              const Text(

                'College Of Engineering Counseling Center',

                style: TextStyle(fontWeight: FontWeight.bold),

              ),

              const Divider(),

              Row(

                mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                children: [

                  const Icon(

                    Icons.email,

                    size: 36,

                    color: Colors.blue,

                  ),

                  const SizedBox(

                    height: 20,

                  ),

                  const Text(

                    'CoE@gamil.com',

                    style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                  ),

                  const SizedBox(

                    width: 20,

                  ),

                  ElevatedButton(

                    onPressed: () {

                      String? encodeQueryParameters(

                          Map<String, String> params) {

                        return params.entries

                            .map((MapEntry<String, String> e) =>

                                '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                            .join('&');

                      }

                      final Uri emailLaunchUri = Uri(

                        scheme: 'mailto',

                        path: emailReceipient[1].toString(),

                        query: encodeQueryParameters(<String, String>{

                          'subject': 'Booking A COUNSELING Appointment',

                        }),

                      );

                      launchUrl(emailLaunchUri);

                    },

                    style:

                        ElevatedButton.styleFrom(backgroundColor: Colors.blue),

                    child: const Text('Book'),

                  ),

                ],

              ),

            ],

          ),

          const Divider(

            height: 5,

          ),

          Column(

            children: [

              const Text(

                'College Of Arts And Natural Resource Counseling Center',

                style: TextStyle(fontWeight: FontWeight.bold),

              ),

              const Divider(),

              Row(

                mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                children: [

                  const Icon(

                    Icons.email,

                    size: 36,

                    color: Colors.blue,

                  ),

                  const SizedBox(

                    height: 20,

                  ),

                  const Text(

                    'CoA&NR@gamil.com',

                    style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                  ),

                  const SizedBox(

                    width: 20,

                  ),

                  ElevatedButton(

                    onPressed: () {

                      String? encodeQueryParameters(

                          Map<String, String> params) {

                        return params.entries

                            .map((MapEntry<String, String> e) =>

                                '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                            .join('&');

                      }

                      final Uri emailLaunchUri = Uri(

                        scheme: 'mailto',

                        path: emailReceipient[2].toString(),

                        query: encodeQueryParameters(<String, String>{

                          'subject': 'Booking A COUNSELING Appointment',

                        }),

                      );

                      launchUrl(emailLaunchUri);

                    },

                    style:

                        ElevatedButton.styleFrom(backgroundColor: Colors.blue),

                    child: const Text('Book'),

                  ),

                ],

              ),

            ],

          ),

          //end,

          const Divider(

            height: 5,

          ),

          Column(

            children: [

              const Text(

                'College Of Arts And Business Environment Counseling Center',

                style: TextStyle(fontWeight: FontWeight.bold),

              ),

              const Divider(),

              Row(

                mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                children: [

                  const Icon(

                    Icons.email,

                    size: 36,

                    color: Colors.blue,

                  ),

                  const SizedBox(

                    height: 20,

                  ),

                  const Text(

                    'CoA&Be@gamil.com',

                    style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                  ),

                  const SizedBox(

                    width: 20,

                  ),

                  ElevatedButton(

                    onPressed: () {

                      String? encodeQueryParameters(

                          Map<String, String> params) {

                        return params.entries

                            .map((MapEntry<String, String> e) =>

                                '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                            .join('&');

                      }

                      final Uri emailLaunchUri = Uri(

                        scheme: 'mailto',

                        path: emailReceipient[3].toString(),

                        query: encodeQueryParameters(<String, String>{

                          'subject': 'Booking A COUNSELING Appointment',

                        }),

                      );

                      launchUrl(emailLaunchUri);

                    },

                    style:

                        ElevatedButton.styleFrom(backgroundColor: Colors.blue),

                    child: const Text('Book'),

                  ),

                ],

              ),

            ],

          ),

          //end,

          const Divider(

            height: 5,

          ),

          Column(

            children: [

              const Text(

                'College Of Health Sciences Counseling Center',

                style: TextStyle(fontWeight: FontWeight.bold),

              ),

              const Divider(),

              Row(

                mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                children: [

                  const Icon(

                    Icons.email,

                    size: 36,

                    color: Colors.blue,

                  ),

                  const SizedBox(

                    height: 20,

                  ),

                  const Text(

                    'COHS@gamil.com',

                    style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                  ),

                  const SizedBox(

                    width: 20,

                  ),

                  ElevatedButton(

                    onPressed: () {

                      String? encodeQueryParameters(

                          Map<String, String> params) {

                        return params.entries

                            .map((MapEntry<String, String> e) =>

                                '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                            .join('&');

                      }

                      final Uri emailLaunchUri = Uri(

                        scheme: 'mailto',

                        path: emailReceipient[4].toString(),

                        query: encodeQueryParameters(<String, String>{

                          'subject': 'Booking A COUNSELING Appointment',

                        }),

                      );

                      launchUrl(emailLaunchUri);

                    },

                    style:

                        ElevatedButton.styleFrom(backgroundColor: Colors.blue),

                    child: const Text('Book'),

                  ),

                ],

              ),

            ],

          ),

          //end,

          const Divider(

            height: 5,

          ),

          Column(

            children: [

              const Text(

                'College Of Health And Social Sciences Counseling Center',

                style: TextStyle(fontWeight: FontWeight.bold),

              ),

              const Divider(),

              Row(

                mainAxisAlignment: MainAxisAlignment.spaceEvenly,

                children: [

                  const Icon(

                    Icons.email,

                    size: 36,

                    color: Colors.blue,

                  ),

                  const SizedBox(

                    height: 20,

                  ),

                  const Text(

                    'COH&SS@gamil.com',

                    style: TextStyle(fontSize: 10, fontWeight: FontWeight.bold),

                  ),

                  const SizedBox(

                    width: 20,

                  ),

                  ElevatedButton(

                    onPressed: () {

                      String? encodeQueryParameters(

                          Map<String, String> params) {

                        return params.entries

                            .map((MapEntry<String, String> e) =>

                                '${Uri.encodeComponent(e.key)}=${Uri.encodeComponent(e.value)}')

                            .join('&');

                      }

                      final Uri emailLaunchUri = Uri(

                        scheme: 'mailto',

                        path: emailReceipient[5].toString(),

                        query: encodeQueryParameters(<String, String>{

                          'subject': 'Booking A COUNSELING Appointment',

                        }),

                      );

                      launchUrl(emailLaunchUri);

                    },

                    style:

                        ElevatedButton.styleFrom(backgroundColor: Colors.blue),

                    child: const Text('Book'),

                  ),

                ],

              ),

            ],

          )

          //end,

        ]),

      ),

    );

  }

}

## algorithm for implementation of database

import 'dart:convert';

import 'package:http/http.dart' as http;

class Auth with ChangeNotifier {

  String \_token = TextEditingController().text;

  DateTime \_expiryDate = DateTime.now();

  String \_userId = TextEditingController().text;

  Timer? \_authTimer;

  bool get isAuth {

    return token.isNotEmpty;

  }

  String get token {

    if (/\* \_expiryDate != null && \*/

        \_expiryDate.isAfter(DateTime.now()) && \_token.isNotEmpty) {

      return \_token;

    }

    return null.toString();

  }

  Future<void> \_authenticate(

      String email, String password, String index, String urlSegment) async {

    const apiKey = 'AIzaSyC9y9njC0xBxbdH8a9oJzV-tHO1mw70yww';

    final url =

        'https://identitytoolkit.googleapis.com/v1/accounts:$urlSegment';

    try {

      final response = await http.post(

        Uri.parse('$url?key=$apiKey'),

        body: json.encode(

          {

            'email': email,

            'password': password,

            'returnSecureToken': true,

          },

        ),

      );

      final responseData = json.decode(response.body);

      if (responseData['error'] != null) {

        throw HttpException(responseData['error']['message']);

      }

      \_token = responseData['idToken'];

      \_userId = responseData['localId'];

      \_expiryDate = DateTime.now().add(

        Duration(

          seconds: int.parse(

            responseData['expireIn'],

          ),

        ),

      );

      \_autoLogout();

      notifyListeners();

    } catch (error) {

      throw error;

    }

  }

  Future<void> signup(String email, String password, String index) async {

    return \_authenticate(email, password, index, 'signUp');

  }

  Future<void> login(String email, String password, String index) async {

    return \_authenticate(email, password, index, 'signInWithPassword');

  }

  void logout() {

    \_token = null.toString();

    \_userId = null.toString();

    \_expiryDate;

    if (\_authTimer != null) {

      \_authTimer?.cancel();

      \_authTimer = null;

    }

    notifyListeners();

  }

  void \_autoLogout() {

    if (\_authTimer != null) {

      \_authTimer?.cancel();

    }

    final timeToExpiry = \_expiryDate.difference(DateTime.now()).inSeconds;

    Timer(Duration(seconds: timeToExpiry), logout);

  }

}

## Testing

Systematic and frequent testing were done to ensure that the system was conforming to and providing the required output. This ensured that the project stayed on course and did no deviate from the specified aims and objectives gotten from the requirements.

## Results

The system is able to provide useful and desired deliverables which corresponds the specified aims and goals.

* Deliverables: The deliverables provided such as the feedback from the system is acceptable and conforms to the aims and objectives set forth.
* Project Performance: The project performance in terms of the desired goods and quality of standard is acceptable.
* Lessons Learned: Many insights were picked in the process of developing The project which will help in future updates of the system.
* The project is able to provide time event flyers and scrollable news to give verified information.
* The chat platform is able to assess the internet and send messages to the counseling center.
* The call plat accesses the phone call logs to pass call to the counseling center.
* The booking center accesses the email function to relay booking information and appointments.
* The audio platform plays the needed audio messages required and approves to students.

# CHAPTER 5

# FINDINGS AND CONCLUSION

## 5.1 Chapter Overview

This chapter discuss about the various finding and conclusion from the project in order to improve upon it in the future. Ways in which these finding can impact a lot of people are listed here with suitable recommendations and references, used in the development and life cycle of the application.

## 5.2 Findings

The project provides the recommended deliverables and output which conform to the requirements set before hand by stakeholders and customers. Some of these findings are:

* The extra packages required to integrate the phone to send and receive email did not agree with my virtual machine and I had to interact directly with the email address by initiating it directly from inside the app.
* The project provides the predesigned deliverables expected from the requirements.
* Waterfall project method worked efficiently than the agile method when developing the application.
* Firebase handles its own database and you can only access it using specially generated token keys.

## Conclusions

In a nutshell, this initiative with its key features, wide range of reach, maneuverability and ease to use is to help ease the stress of freshers on campus, aid continuing students with their daily lives on campus and help KCC reach out to the student populace. My aim is to bridge the long invincible gap that exist between KNUST student populace and the KNUST COUNSELLING CENTER by using my healthcare app initiative which has both parties in mind without the stress of fear, insecurity, shyness, influence from fellow peers, etc. getting in the way of the motives of KCC reaching out to students.

## Challenges/ limitations of the system

* The app will be limited due to network issues based on the geographical location of the students. Due to the undeniable fact that students would need internet bundle to access the app, bad connections or low and no internet bundle time will make accessing the app impossible.
* Aside students, lectures who seek to use the app for their benefits will not be allowed to access the app.
* Graduates who have completed their term of enrollment in the school and also teaching assistance will not be permitted to use the app.
* Unable to conduct video calls with counselor.
* Delay in message replies due to the busy scheduled of the counselors and peer counselors.

## Lesson learnt

* Virtual machines are slower when running test with it but physical mobile devices with developer option approved works faster.
* Most flutter codes do not work on newer virtual machines unless flutter SDK has been updated.
* Codes written with older versions of flutter and dart do not work with integration of new flutter elements.

## Recommendations for future works

* Inclusion of a video calling function to include a face-to-face interaction.
* Inclusion of lecturers, master’s student and postgraduate students to sign up and use the app.
* Inclusion of peer counselors on the app to increase the counseling centers reach.

## Reference

Flutter Documentation. (2023). http://www.pub.dev

MJSD, CODING. (2023, APRIL). [www.youtube.com](http://www.youtube.com)

Schwarzmuller, M. (2016). Flutter and Dart. http://www.udemy.com