STRESS MENTAL HEALTH SYMPTOM ASSESSMENT MOBILE APPLICATION FOR YOUNG ADULTS

BY

LEE CHUN HOONG

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Stress Mental Health Symptom Assessment Mobile Application for Young Adults

Academic Session: June 2023

I LEE CHUN HOONG
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

ChunHoong

(Author's signature)

(Supervisor's signature)

Address:
114, Jalan Indah 1/25, Taman
Bukit Indah, 79100 Iskandar
Puteri, Johor Bahru, Johor.

Dr. Mogana a/p Vadiveloo
Supervisor's name

Date: 15/09/2023

Date: 15/09/2023

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: __15/09/2023____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ______***Lee Chun Hoong***________ (ID No: __***1905878***_) has completed this final year project entitled "Stress Mental Health Symptom Assessment Mobile Application for Young Adults" under the supervision of __Dr. Mogana a/p Vadiveloo______ (Supervisor) from the Department of __Computer Science______, Faculty of _Information and Communication Technology__ .

I understand that University will upload softcopy of my final year project / dissertation/ thesis in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

ChunHoong

(*LEE CHUN HOONG*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Stress Mental Health Symptom Assessment Mobile Application for Young Adults**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ChunHoong

Name : LEE CHUN HOONG

Date : 15/09/2023

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr Mogana a/p Vadiveloo for all the guidance that she provided me for the project. It has allowed me to be exposed to the skills and techniques which are related to application development.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Mobile health applications, better known as “mHealth” applications, are getting popular nowadays. Mobile digital health technology enhances patient care by improving condition monitoring and diagnosis methods, resulting in more timely and comprehensive care. In this fast-paced world, young adults are getting stressed compared to previous years as they experience more difficulties than before in terms of work, school, and relationships. Stress harms health and productivity, whether a constant struggle or an occasional flare-up. This project is to study the development of a stress mental health symptom assessment mobile application to help people, especially young adults or university students, measuring their stress levels and then carry out appropriate activities for relieving stress. The proposed application intends to solve several limitations found in the existing stress management applications in the mobile applications market. Some existing applications such as Headspace and Smilin Mind do not have a proper stress level assessment backed by solid scientific studies. Therefore, this proposed **stress mental health symptom assessment mobile** application can allow users to conduct a stress assessment session in the application to generate a report of the stress level. It is developed using **Android Studio, React Native, Android SDK, Javascript, Google Firebase, and an Android smartphone.** A series of practical activities are recommended to the users to relieve the stress. Moreover, users can provide more inputs based on their feelings throughout the day. One of the primary functionalities of the application is to incorporate a machine learning algorithm which is K-Nearest Neighbor (KNN) classification technique for panic attack prediction feature to enhance the emotional identification and offering users an artificial intelligence (AI) chatbot. As for the panic attack prediction feature, the application will transmit user-input responses to a model previously trained using historical data. Before this stage, the KNN algorithm is employed to construct the model using Google Form response data as part of the application development process. This model will then be integrated into the React Native application for use. It will gather information such as the user’s gender, age, current course of study, current year of study, marital status, and any previous instances of seeking special treatment. Subsequently, employing the K-Nearest Neighbors (KNN) algorithm, the model shall forecast the likelihood of experiencing a future panic attack. Remarkably, the KNN model exhibits a testing accuracy of 70.37%, signifying a commendable outcome from

both the training and testing phases. This information can empower them to take necessary steps for preparation or prevention if a panic attack is anticipated in the future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AI	Artificial Intelligence
API	Application Programming Interface
CASE	Computer Aided Software Engineering
CGPA	Cumulative Grade Point Average
DASS-21	Depression, Anxiety, Stress Scale - 21
GPT	Generative Pre-Trained Transformer
IDE	Integrated Development Environment
KNN	K-Nearest Neighbour
MBSR	Mindfulness Base Stress Reduction
RAM	Random Access Memory
SDK	Software Development Kit
SDLC	Software Development Life Cycle

CHAPTER 1: Introduction

1.1 Background Information

Nowadays, mostly every people are living a fast-paced life to keep up with the rapid development of the society. Most people are focused on maintaining the physical health, instead of putting the same effort in ensuring the well-being of the mental health. This lifestyle causes them often to overlook their mental well-being, and thus lead to accumulation of stress level. Numerous causes, including pressure at work, marital stress, financial difficulties, and academic difficulties, can lead to stress [28]. Neglecting one person's stress level can result in a loss in general mental and physical health and may even trigger additional mental health problems [29]. For instance, in secondary and higher education settings, students deal with a variety of persistent normative stresses, which are described as everyday difficulties including continuing academic responsibilities [26]. Approaching psychologists and psychiatrists for help in coping with the adverse feelings and effects brought on by stress is the most typical way for dealing with mental health concerns. However due to the limited number of certified mental health professionals and the increased demand of the psychological practitioners especially during the pandemic [27], many individuals have gradually failed to improve the mental health, and further elevating the stress. When the stress level is severe, it can lead to imbalances in emotions and thoughts, hindering a person's positive outlook.

The emergence of mobile health applications changed this scenario. People can now download applications on their phones to understand their mental health and gradually alleviate stress [8]. Such applications have become popular in the mobile app market, encouraging many to download and confront their issues. These applications provide effective solutions to reduce stress, particularly through practices like meditation, which helps individuals organize their thoughts and find calmness.

1.2 Problem Statement and Motivation

Better known as mHealth applications, mobile health applications have been developed and published in Google Play Store or App Store. These applications, such as Headspace, Shine, Amaha (Innerhour), and Smiling Mind, aim to promote physical and mental health changes among users. As interest in mental health has grown, so have demands for high-quality care. The demand for mental healthcare via mobile communications and social networking technology has surged in reaction to this situation [8]. In terms of mental health applications to tackle the issue of depression, anxiety and stress, these applications help users to save time to evaluate their mental health in a short time rather than spending time at a psychological clinic if the users are only experiencing mild conditions. Besides, mental health applications bring simple interventions to behavioural health changes among the users by minimizing the risk factors. Therefore, applications seem to be the perfect delivery platform for both straightforward and efficient solutions. Several researchers have studied the efficiency of mobile phone applications in changing health-related behaviour across various mental health conditions. They have proved to help overcome mental health issues among the participants [9, 10, 11].

In recent years, there has been increasing stress among Malaysians, especially when they prioritize their physical health more than mental health. When statistics show a 29% increase in instances among Malaysians who feel stress compared to the year 2011, it is clear that the mental health issues, including stress, depression, and anxiety, have substantially increased in recent years [6]. Stress is a mental, emotional, or bodily tension brought on by internal and external factors, like worry and excessive effort [7].

As the number of users who should use the mental health application has been increasing steadily, especially the university students, some limitations arise. Most people live in a fast-paced environment, and the heavy workloads or school homework might affect their mental health [23]. Unfortunately, some people do not realize the importance of maintaining good mental health. Therefore, a mobile application could be beneficial to assist people in evaluating their stress levels and solving the issues efficiently.

Many mental health applications in the mobile application market, especially mindfulness-based applications such as Headspace, Shine, Amaha (Innerhour), and Smiling Mind [22], encourage the need for meditation to manage stress. However, the applications lack relevant stress assessment with a solid scientific background. Therefore, the users should be able to measure their well-being before participating in a registered mindfulness program. This is because both the application and users can understand the issues better to tackle the mental health issue more effectively. Furthermore, the application should continuously evaluate users’ mental health to track their progress when carrying out the activities and whether their well-being is improving or not.

Moreover, a few mental health applications, for instance, Headspace and Shine [24], do not provide much user flexibility when using the application. Besides providing the most suitable mindfulness-based program based on different user requirements, another crucial part of these applications is ensuring that users can track their thoughts along the way they use the apps. For example, any negative thoughts should be recorded as personal notes or a personalized mood tracker. These components are considered critical in weighing users' stress level scores. When more input features are available in the application, it can produce a more accurate and consistent score report for the users.

The aim of this proposed application is to help users to understand their stress issue by evaluating their stress level from the application. Even though it is more effective and advisable to seek professional help outside, this proposed application is still feasible in helping those with stress mental health concerns. When the users conduct a stress evaluation with the application, it benefits them because excessive time consumption is avoided, especially when they have a busy schedule. Besides conducting the stress evaluation session, the proposed application will concentrate on collecting more input features from the users, including the thoughts and mood of the day. Based on various inputs provided by the users in the application, the system can conclude the users' stress mental health. Besides that, the application aims to deliver a sense of reliability to the users by generating a detailed analysis report after each evaluation. Thus, users get a deeper insight in terms of their symptoms that may possibly correlate to stress mental health issues. Moreover, the proposed application focuses on generating a string of suggested stress relieving activities that attempts to ease the user's mental condition. In addition, these activities can be classified into three categories: meditation, music relaxation, and physical activities. Furthermore, the proposed application utilizes a machine learning algorithm to create a panic attack prediction function driven by user input, with a focus on engaging students in the classification process. Additionally, a modest artificial intelligence (AI) chatbot is seamlessly integrated into the application, offering users a platform to communicate and express their emotions when they find themselves without someone to confide in.

To summarize, the motivation of the proposed application will increase the users' awareness on their stress mental health.

1.3 Objectives

The application developed must achieve the following objectives to ease the users to use the proposed stress management application.

1. To develop an android-based mobile application that can provide stress assessment sessions to the users and ensure they assess before selecting the program.

Most existing mHealth applications in the current app market fail to provide a comprehensive stress assessment session to the users before they access to the options in the application to improve their mental health. This is not helpful for the target users especially the users wish to fully understand their mental condition during the period. This issue with reduce the effectiveness of the application and discourage the users to be in an active role to manage their stress efficiently. When users are not able to understand their stress value or scores, they will lose direction and might cause them to choose the ineffective program or options which is not tailored to their needs to tackle stress. Therefore, to address this issue, the proposed application will provide a short and simple stress assessment session so users can take part into the assessment at least once especially first-time login user to obtain accurate result to increase the reliability of the application.

2. To generate a report after the stress assessment along with a detailed analysis and description of users' stress levels.

After the user has completed the DASS-21 questionnaires related to stress issues faced by the users, they shall receive a full report which consists of a detailed analysis based on the stress score categories. There are five categories of the stress score value ranging from Normal to Extremely Severe, depends on the scoring obtained during the evaluation. Users will receive different reports based on their scoring at the time and the reports will includes the causes of different stress levels, which explains their stressors and allow them to understand the impact of stress on their daily lives. Besides that, this proposed application is aimed to help the users to provide personalized stress tackling strategies by including some helpful recommendations to address the stress issues of different categories.

3. To provide a mobile application with a list of programs or activities that can help users reduce stress.

This proposed mHealth application is to allow users to access to variety options of programs or activities to manage their stress. Mindfulness Based Stress Reduction (MBSR) programs was adapted as the primary channel for the users to reduce their stress. It has been proved of its effectiveness in reducing stress through mindfulness practices. However, MBSR programs is originally a course which requires number of weeks to complete and able to achieve mindfulness. The traditional way is not feasible for the users who are searching for quick solution for their stress issues. Since the application is to help users to tackle with their stress with minimal time required, the MBSR programs was categorized into beginner, intermediate, and advanced levels. The functions developed can help users to choose the correct activities with appropriate difficulty levels to achieve the best result in the shortest time possible based on their interest level. Other than providing MBSR programs, the application has one of the objectives to help users to understand themselves better or embrace their inner thoughts. Therefore, the application will provide other functions such as mood recording functions and note-taking sections for the users to record their daily experiences and save them for self-reflection, providing a better understanding of their mental health and emotional well-being.

4. To provide a function to keep track of the users' progress while undergoing the Mindfulness Based Stress Reduction program (MBSR).

Some of the existing applications lack a feature to track users' progress when carrying out the activities to relieve their stress. This problem will lead to a confusion among the users to fully aware and remember the progress they have made in the activities. Especially when users choose to carry out MBSR program, keeping track of the progress is important for the users to learn how to control their timing on each activity before proceeding to the next one. Hence, a progress tracker will be developed inside every MBSR programs specifically for the meditation exercises. As users listen to the audio instruction to carry out the activities, there will be checked buttons on every instruction to record their progress during the meditation. They can pause the meditation exercise and mark the instructions where they currently completed and able to resume later. This function helps to provide users with more flexibility and reduce the pressure to complete the meditation in one sitting.

5. To provide an integrated chatbot feature inside the application to encourage users to express their feelings or share their stories of the day.

An integrated chatbot function will be included to the proposed mHealth application to help it achieve its goal by offering a comprehensive solution for stress management. The chatbot will act as the users' personal assistant, encouraging them to share their daily experiences or convey their emotions. The chatbot will engage the users in the conversation, helping them to reduce feelings of isolation and anxiety. In addition to making users feel heard and understood, this feature will recommend appropriate actions from the application to reduce their stress based on their current circumstances, thanks to its natural language processing (NLP) capabilities. For example, the chatbot might suggest a meditation exercise or breathing technique to make them feel relax. Users will benefit from a supportive and interactive experience thanks to the chatbot, which will make the process of managing stress more enjoyable and successful.

6. To implement a machine learning algorithm to enables users to predict future outcomes and utilize a language model.

The purpose of this investigation is to apply K-Nearest Neighbor (KNN), a machine learning algorithm to empower users with a more profound insight into their stress levels, serving as an alert system for potential panic attacks in the future. In pursuit of enhancing the model's performance and precision, various types of classification algorithms have been studied and explored for best accuracy. This exploration includes the thorough investigation of algorithms such as K-Nearest Neighbor (KNN), Random Forest, Decision Tree, and XGBoost. In summary, the classification algorithm chosen will generate an accurate result whether a user might encounter a panic attack based on their academic background. Consequently, users will be better equipped to prepare for and reduce the possibility of such occurrences.

7. To provide a user-friendly interface to ease the communication between the user and the application.

User interaction is the main focal point in determining the success of mHealth application as it relies on the user's input to evaluate stress. This application emphasizes user-friendly interface and aesthetically soothing layout, allowing users to seamlessly express their feelings without any confusion in usability. To enhance the application's reliability, users can easily access all the features without any complicated process while providing feedback though a simple interface. Thus, users may be more likely to continuously utilize this application which in turn may help to improve their stress management.

1.4 Project Scope and Direction

The proposed mobile application does not aim to treat mental disorders of any forms including the severe ones such as depression and is instead intended to address the stress mental health issues among the young adults. The stress management application is categorized as one of the Mobile Health applications. Therefore, the proposed stress management application must fulfil most user requirements and user-friendliness when they wish to reduce their stress via a mobile application. The application's functionality starts to be represented when the users conduct stress level assessments. The proposed application is developed to avoid time-consuming among the users when undergoing the assessment because the assessment is based on the DASS-21 questionnaire [2]. DASS-21 is chosen as the channel to obtain a response from the users when assessing because the effectiveness of this questionnaire has been studied and proved with scientific study, where it can identify the users who are currently under stress. Besides that, the proposed application is integrated with a complete mindfulness-based stress reduction program, in which the idea of MBSR [1] is adopted to assist users in tackling stress. A report is generated to the users upon the completion of the assessment.

Furthermore, the assessment session, users will receive the recommendation of activities if they are unfamiliar with or unsure which program to select. Moreover, the proposed application will have an integrated chatbot which provides several essential functions to communicate with users. By integrating a machine learning algorithm to forecast potential outcomes within the proposed application, a binary classification model will be created. This model will gather user input regarding their existing academic background and qualifications, with the aim of identifying and assessing the likelihood of a user encountering a panic attack attributed to accumulated stress. The proposed application will also integrate React Native Expo Audio and Video Library to allow users to play relaxing music playlists while watching soothing video to help manage their stress level in the application. In short, the help of stress assessment questionnaire, analysis report, progress bar, integrated chat bot, machine learning classification model, recommendation of activities to relieve stress, and user-friendly interface on the mobile application help users manage their stress more efficiently.

1.5 Contribution

The main contribution of this work is to implement a stress management system via a mobile application. It is expected that users can understand their stress level and identify the root problem of their stress level by a more efficient way of conducting stress assessment. Users can also be more confident in the analysis report generated and choose to carry out the Mindfulness Based Stress Reduction (MBSR) program later. Moreover, with different functionalities implemented inside the application, users are expected to achieve better results in reducing stress. In addition, it is known to be user-friendly to ease the user's use of the application. Not only that, the incorporation of a machine learning model into the proposed application, designed to be installed on mobile devices, will enable users to directly experience the efficiency of utilizing artificial intelligence for managing stress. In this era where everyone is familiar with technology devices, a good stress management application is necessary and beneficial to ease users resolve their stress issues.

1.6 Report Organization

The details of this development are shown in the following chapters. In Chapter 2, the effectiveness of DASS-21 was studied to ensure its usefulness to the target users of the proposed application. Besides that, there were a few of the existing stress management applications to be selected and reviewed in Chapter 2. Critical analysis will be investigated to compare between the applications' strengths and weaknesses. The system methodology, system requirement and system design diagram are discussed in Chapter 3. Chapter 4 will be discussing about the preliminary work of this project, such as hardware and software setup, and the preliminary prototype of the application. Finally, this report will be concluded with Chapter 5.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Previous researchers have studied the effectiveness of mental health applications on human mental health improvement. In addition, mental health applications are highly acceptable among students because mobile phones integrate into university community members' life as daily drivers. Therefore, this chapter will focus on the minimal time commitment to mobile mental health applications to generate a detailed mental health summary report for users. Other than that, functionalities of the existing stress management application, such as stress level assessment, the solution to different stress levels among the students, and user experience in the applications are discussed.

2.2 Evaluation of Depression Anxiety Stress Scale Test 21 (DASS-21)

Some significant and widespread psychological studies in the existing literature have summarised using human psychological signals [1] to detect and evaluate stress levels. The signals include electrocardiogram, blood pressure, skin temperature, heart rate, breathing rate, and respiratory volume. However, this is time-consuming for some patients as they are usually busy and request a brief assessment with a precise diagnosis. Therefore, several scales and questionnaires were developed to provide patients with proper and relevant assessment tools. However, fewer scales can be used to measure the stress levels. One of the commonly-used questionnaires is DASS-21. DASS was introduced in 1995, and the shortened version, DASS-21, is developed and used to assess depression, anxiety and stress levels among people in a shorter assessment time. The questionnaire consists of 21 questions and yields three subscale scores for depression, anxiety, and tension/stress [2]. Patients need to answer all 21 questions. Accordingly, the answers are based on a scaled range, from 0 (Did not apply to me at all) to 5 (Applied to me very much, or most of the time). The scores are then calculated separately based on three subscales to evaluate the mental health levels of the patient.

The reliability of DASS-21 is tested in Vietnam [3] and Turkish [4]. The result is that the scale test is highly acceptable among the participants and was proven to help detect mental health levels. Le et al. [3] stated that the correlation r = 0.90; 0.80 and 0.79 represented the correlations between the General Distress factor and Depression, Stress, and Anxiety. Besides that, the correlations between these three subscales are reduced. Substantial evidence is shown to support the statement that while these three components have distinct characteristics, they also represent a general distress component.

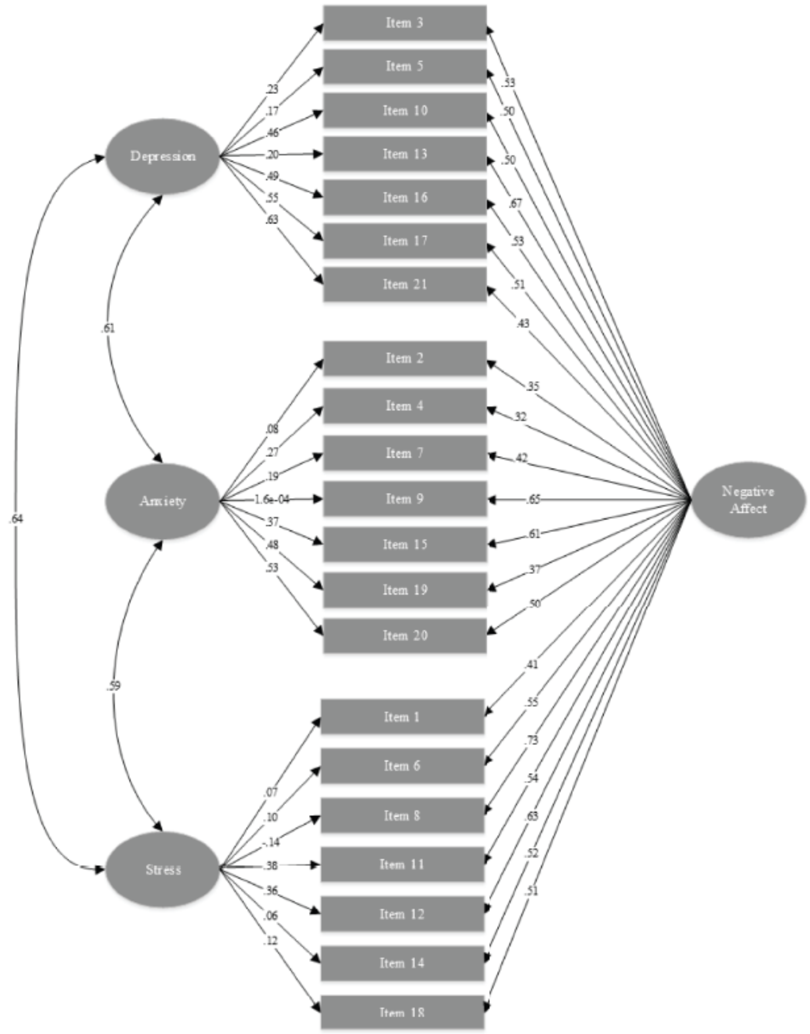


Figure 2.1 Factor loading of DASS-21 [3]

Although the results show a good correlation between the three subscales, the factor loading diagram above stated that the Stress factor has only two significant items. Item loadings of the Depression factor were better because it ranges from 0.17 to 0.63. Regarding the Anxiety factor, except for item 9 (I was worried about situations in which I might panic and make a fool of self), ranging from 0.081 to 0.53. Compared to Depression and Anxiety Factor, the Stress factor has the least number of significant items: item 11 (I found myself getting agitated) and item 12 (I found it difficult to relax). According to Le et al. [3], various reasons contributed to this issue. Other items in the Stress factor are significantly small or even negative values. This is because the questionnaire has been translated from English to Vietnamese. The translation was misinterpreted and unable to translate the item’s original meaning. Thus, the translation was not accurate enough and may thus affect the results. For example, item 18 (I felt that I was sensitive) in the Stress factor failed to convey the meaning to “easily move to anger”. However, the results returned from Turki [4] were different from the test in Vietnam. It explained the difference in age of the participants from different places, and the emotional development state has not reached a similar stage yet. Besides social characteristics and cultural differences, translating the questionnaire from English to Vietnamese may also contribute to these differences.

Cronbach’s alpha measures internal consistency. This describes how closely a group of things are collectively related. It is also known as the scale reliability measure [5]. Typically, Cronbach’s alpha is expressed on a scale from 0 to 1, with higher values signifying higher dependability. It is a crucial idea utilized in the examination and assessment of questionnaires. For example, Sariçam [4] mentioned a = 0.81 the Cronbach alpha internal consistency coefficient for the stress subscale, a = 0.87 depression subscale, and a = 0.85 anxiety subscale.

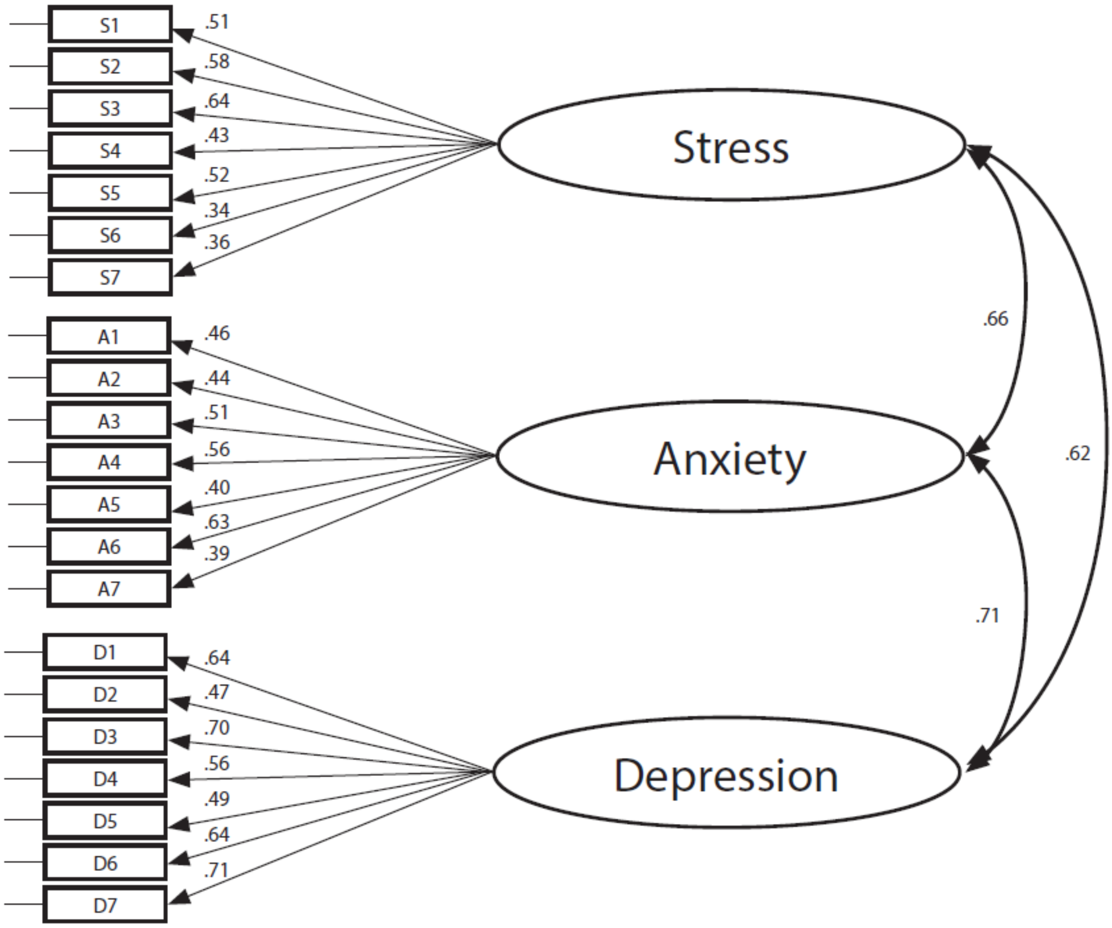


Figure 2.2 Confirmatory factor analysis of DASS-21 [4]

Figure 2.2 shows that the item factor loads vary from 0.34 to 0.64 for the stress subscale, from 0.39 to 0.63 for anxiety, and from 0.47 to 0.71 for depression. According to the abovementioned standards, it is possible to assert that the sample group of Turkish university students protects the scale's triple structure. These values have proven that DASS-21 is suitable for the development and adaptation criteria. The DASS-21 summarises the stress, anxiety, and depression levels in both healthy people and patients. It has been proposed that it may also present a chance for diagnosing and assessing problems and carrying out psychiatric, psychotherapeutic, and counselling applications.

To be more precise, DASS-21 contains the limitations of misinterpretation of the original meaning of the questionnaire items. When people have difficulty understanding the questions, thus it will cause a decline in the reliability of DASS-21. Therefore, using the questionnaire to measure the students' stress scores should be used with caution if reconstructing the questions is required. Moreover, DASS-21 should be performed by considering a more extensive set of clinical samples to produce more accurate reliability and consistency score. Finally, it would be good to confirm the DASS-21's validity for usage among students in future studies by comparing it to a standard diagnostic tool for mental disorders.

2.3 Evaluation of Existing Similar System

2.3.1 Headspace

Headspace [12] is an application that not only tackles the stress issue among the users but also helps reduce depression thoughts and most negative feelings. The application provides meditation courses and relaxing music to help manage stress, sleep, and anxiety. In addition, it aims to practice mindfulness in daily life. Headspace's main features are guided meditation designed by a professional yoga instructor, sleep meditation, and breathing exercises. These meditation exercises encourage users to meditate for different experience levels and lifestyles. In addition, users can choose to meditate with friends or carry out 3-minute sessions or longer meditations to set their intentions a day [18].

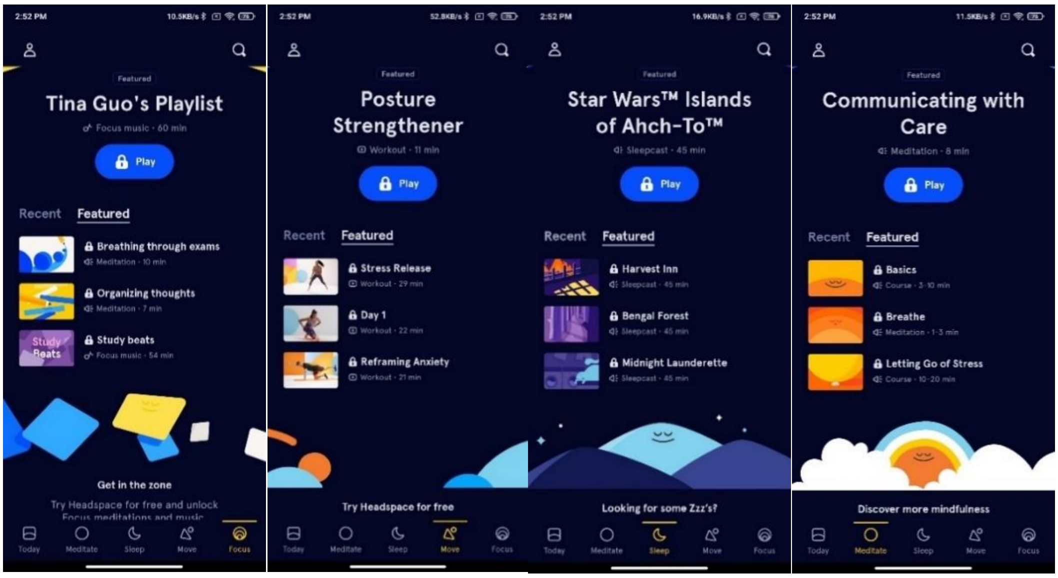


Figure 2.3 Different mindfulness-based features provided in Headspace

According to Figure 2.3, there are four main functions provided in Headspace to assist the user in relieving negative feelings. These features include guided meditations and hundreds of mindfulness exercises, daily meditation practice on a new topic each day, sleep meditations with calming stories paired with relaxing sounds, and "move mode", which is mood-boosting and motivating at home workouts to relieve stress and release everyday tension, and “focus mode” to boost concentration and productivity with focus-enhancing music. To enjoy all the features locked in the free version, users need to subscribe to the membership to explore the entire Headspace library, including all access to hundreds of meditations and mindfulness training.

Each main feature consists of more subcategories designed according to different user needs. Users will explore more in each feature to find the one suitable for them. For example, users can choose whether or not to train to join meditation sessions with a professional meditation teacher. The session is usually 10 minutes long and very suitable for users with a busy schedule. Besides that, an “SOS function” is designed to help users quickly relieve negative feelings, including losing their temper, feeling overwhelmed, burning out, panicking, in pain, or flustered. This “SOS function” is helpful when users need a quick time to take a break. Suppose users have any doubts or difficulties in practising mindfulness in their daily life [13]. In that case, they can approach a special session in Headspace to learn mindfulness tips to deepen the practice and thus find answers to their mindfulness questions. In short, the application covers most mindfulness aspects to ensure users can be kind to the mind.

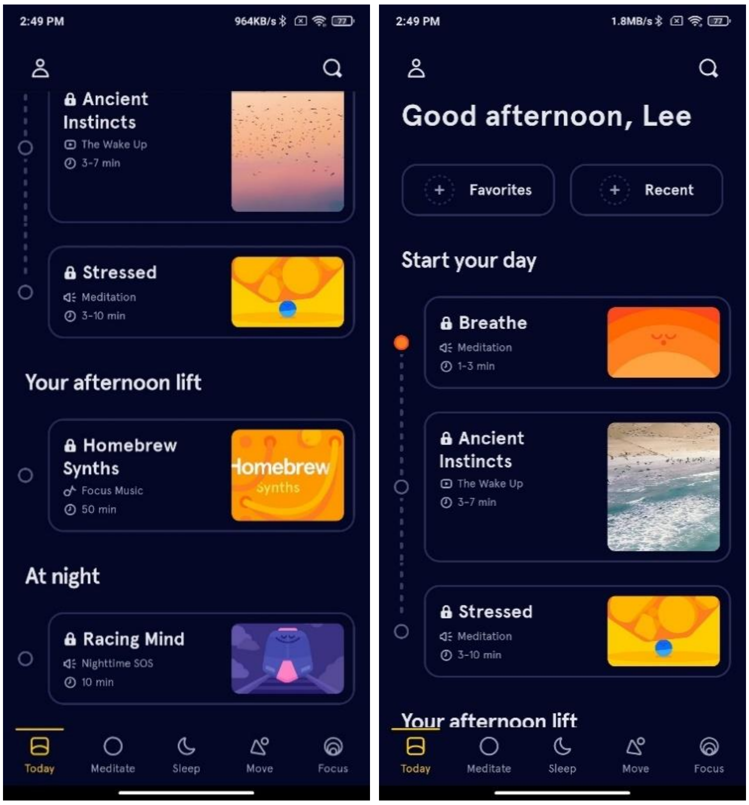
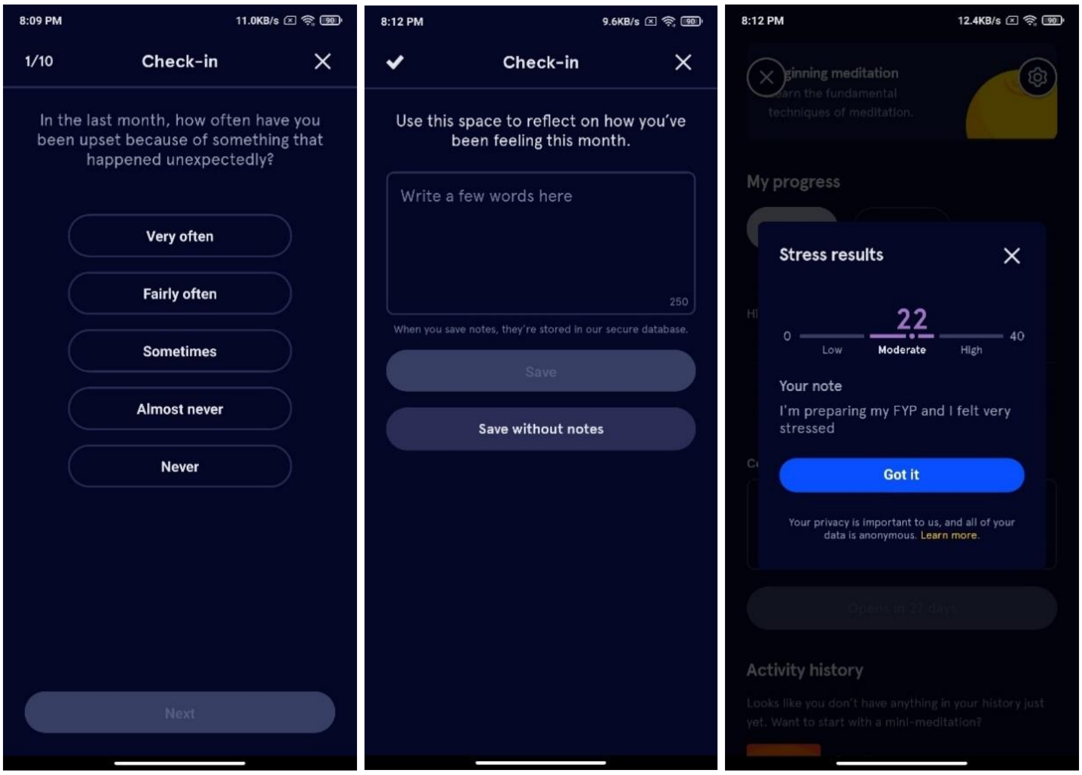


Figure 2.4 Home page of Headspace

The homepage of Headspace is shown in figure 2.4. Even if the users cannot decide which training courses to carry out due to the many subcategories in each feature, the application recommends the activity from morning to night. This series of recommendations encourages users to practice mindfulness as the activities include simple meditation, breathing activities and soothing ambient sound. Users who meditate can download the session for future offline use if they find it interesting and valuable. When using the application, users will also get notifications of meditation reminders, mindful moments, bedtime, and wake-up reminders [12].

According to Figure 2.5 below, a stress assessment session in Headspace allows users to measure their stress levels. Users only have to answer ten simple questionnaires by reflecting on their moods and thoughts for the last month. The answers range from "never" to "very often" for every ten questions. Moreover, the application will prepare a text box to ask users if they have any extra notes to record how they feel in the last month. The notes can be viewed after users submit the stress assessment to generate stress results. However, each user can only carry out one stress assessment per month, making it difficult for users to determine their stress level only once.



In the last month, how often have you been upset because of something that happened unexpectedly?

Use this space to reflect on how you've been feeling this month.

When you save notes, they're stored in our secure database.

Your note

I'm preparing my FYP and I felt very stressed

Your privacy is important to us, and all of your data is anonymous. Learn more.

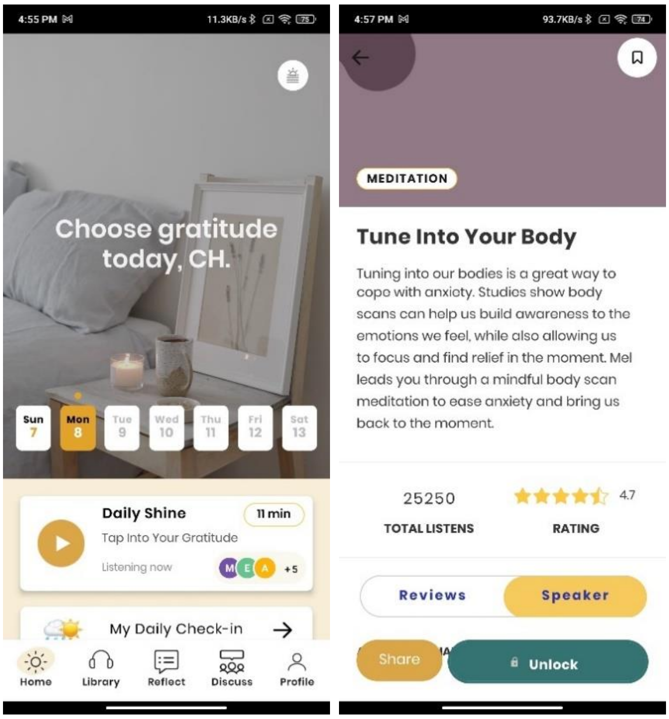
Looks like you don't have anything in your history just yet. Want to start with a mini-meditation?

Figure 2.5 Stress assessment in Headspace

The advantage of this application is that it provides a wide range of mindfulness-based meditation activities to help different users overcome stress and anxiety. In addition, users can explore mindfulness practices from different aspects, such as breathing exercises, focus music and sleep meditation. However, this application's downside is that users are only allowed to perform stress assessments each month. This disadvantage will lead to users not being able to recall all the scenarios in the past month.

2.3.2 Shine

Similar to Headspace, Shine is an application which provides meditation classes to users to calm anxiety and stress [15]. It is also a paid application which requires users to subscribe to a monthly plan. Therefore, users can unlock the complete mental wellness program when they become the Shine Premium member. This application is featured as the 2019 Google Play Award Standout Well-being App and the 2019 Webby Award Winner for Best Lifestyle App [14]. One of the winning reasons is that this application is packed with different meditation programs, including thousands of meditation libraries. The meditation library is created by experts and customized with personalized tools and content based on user needs.

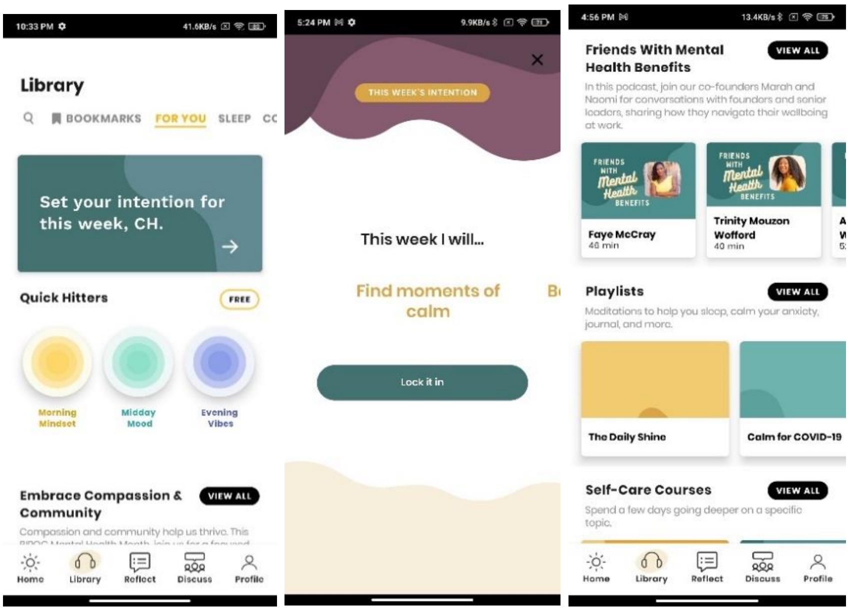


Tune Into Your Body

Tuning into our bodies is a great way to cope with anxiety. Studies show body scans can help us build awareness to the emotions we feel, while also allowing us to focus and find relief in the moment. Mel leads you through a mindful body scan meditation to ease anxiety and bring us back to the moment.

Figure 2.6 Home page of Shine application

Based on Figure 2.6 above, users can access the daily meditation plan when they log in to the application. This feature allows users to customize the meditation library, plan their community care events, and track their daily moods. If users select the meditation plan on the homepage, they can read a brief introduction to the meditation session. Users can also acknowledge the total number of listening to the particular meditation class and the overall rating and reviews posted by the other users. This petite engagement helps users be less worried about the program's effectiveness and ensures that the meditation is effective and can provide quality content. If users find this meditation class helpful, they can share it with their friends or bookmark the class for future quick access [15].



Set your intention for this week, CH.

Compassion and community help us thrive. This

This week I will...

Find moments of calm

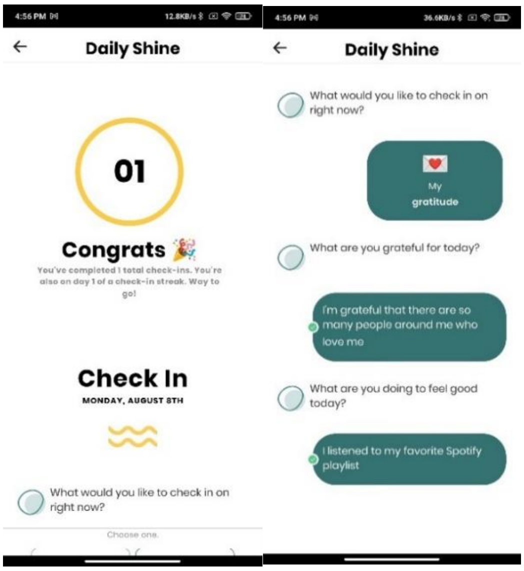
In this podcast, join our co-founders Marah and Naomi for conversations with founders and senior leaders, sharing how they navigate their wellbeing at work.

Meditations to help you sleep, calm your anxiety, journal, and more.

Spend a few days going deeper on a specific topic.

Figure 2.7 Search engine of Shine application

Figure 2.7 above shows the search engine developed in the Shine application. There are various searching methods in Shine to help users clearly define the correct specification of the meditation plan. Users can search their meditation library by filtering the meditation into different purposes such as sleep, stress and anxiety, work, sounds, and confidence. Besides that, users can set their intention for a week to search for a relevant meditation class. This indicates that the users can set the intentions and act as a meditation purpose so that the application will add meditation classes to the library. Moreover, the meditation library can be filtered by suitable period from morning to evening [15]. In brief, the application provides a user-friendly searching function to allow users to find the most suitable meditation class.



You've completed 1 total check-ins. You're also on day 1 of a check-in streak. Way to go!

What would you like to check in on right now?

What are you grateful for today?

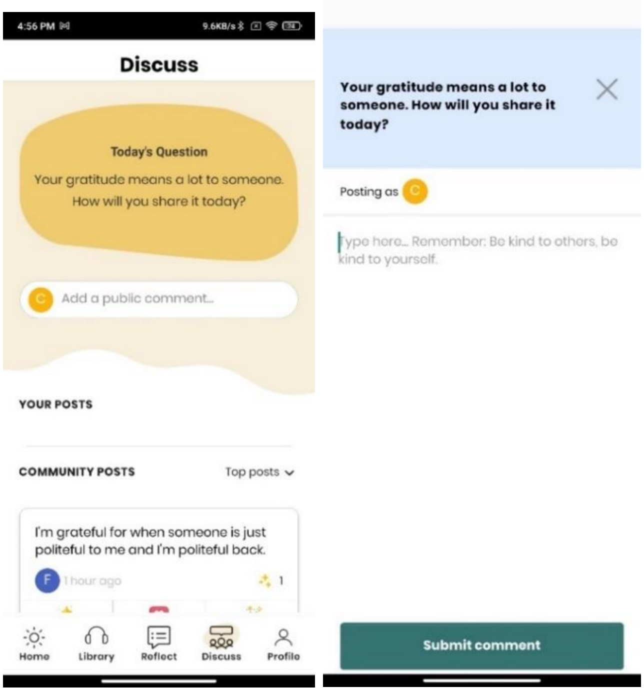
I'm grateful that there are so many people around me who love me

What are you doing to feel good today?

I listened to my favorite Spotify playlist

Figure 2.8 Gratitude check-in function

The other feature provided in Shine is the gratitude check-in function, as shown in Figure 2.8. Users can record any gratitude feeling known as "Daily Shine" in the application on that day. If users experience anything that makes them feel grateful or things they have done to make them feel good, they can record them in the "Daily Shine" section. Therefore, users can check back the check-in record in future. However, users cannot express all kinds of feelings or negative thoughts because this function only allows users to record gratitude.



Your gratitude means a lot to someone. How will you share it today?

I'm grateful for when someone is just politeful to me and I'm politeful back.

Type here... Remember: Be kind to others, be kind to yourself.

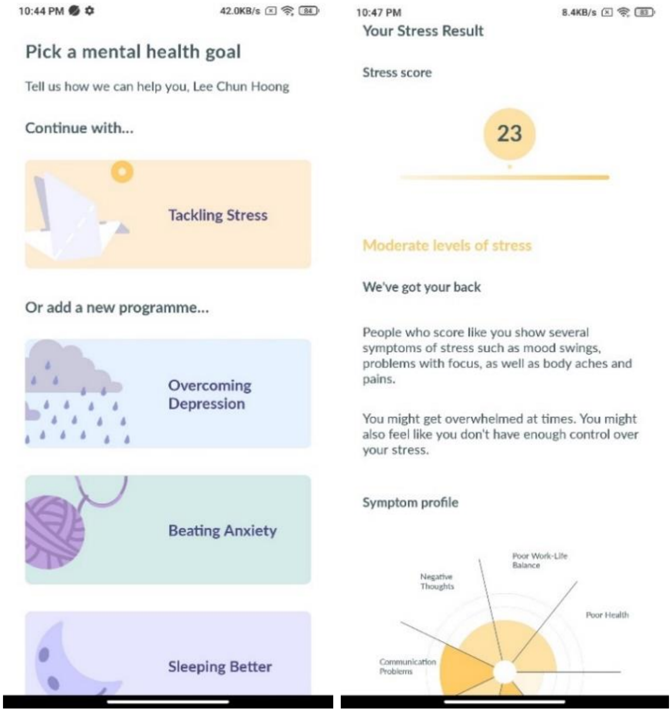
Figure 2.9 Discussion room in Shine application

In addition, the Shine application also encourages users to join the community to share their sincere thoughts on the positive side, as illustrated in Figure 2.9. There are different questions posted in the discussion room every day. Users should post any public comment relevant to the question and share it with other users to spread positive feelings among the community. Therefore, this function gathers the users together to interact with each other and fight against negative thoughts [15].

In a nutshell, the application has the advantage of providing a wide variety of searching options so users can find the best meditation class for themselves. Furthermore, the application has an online discussion community to share positive thoughts. However, the application has no stress or negative thoughts assessment section. Users can only reflect on themselves using the check-in function, but it cannot track anger, stress, or anxiety.

2.3.3 Amaha (InnerHour)

Amaha, also previously known as InnerHour, is developed to boost mental health and fight depression and stress [16]. Users in Amaha will get access to self-care, therapy, and community support and this application are built by trained psychologists and licensed psychiatrists. It aims to help users struggling with depression, stress, anxiety, and sleep.



Pick a mental health goal

Tell us how we can help you, Lee Chun Hoong

Continue with...

Or add a new programme...

Your Stress Result

Stress score

23

Moderate levels of stress

We've got your back

People who score like you show several symptoms of stress such as mood swings, problems with focus, as well as body aches and pains.

You might get overwhelmed at times. You might also feel like you don't have enough control over your stress.

Symptom profile

Figure 2.10 Choosing mental health goals and developing stress result

Based on Figure 2.10 above, users who successfully registered an account in Amaha will be redirected to select the mental health goal to be achieved. Note that each user can choose more than one mental health goal in the application. The six specialized mental health courses include tackling stress, overcoming depression, beating anxiety, and sleeping better [16]. For example, if users select tackling stress as the goal, they need to take a stress assessment to generate a stress result. From the stress result generated from the application, users will understand their stress score with a detailed explanation and graph. However, users cannot remove a programme from the application because it will consider that users are still under the programme all the time. Every six mental health courses offer self-care activities such as affirmations, guided journals and anxiety relief meditation audios. The courses are based on the principles of mindfulness and positive psychology.

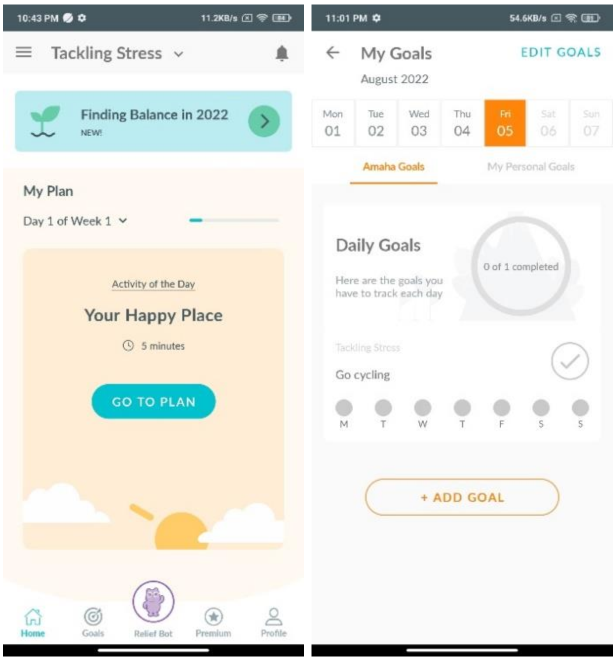
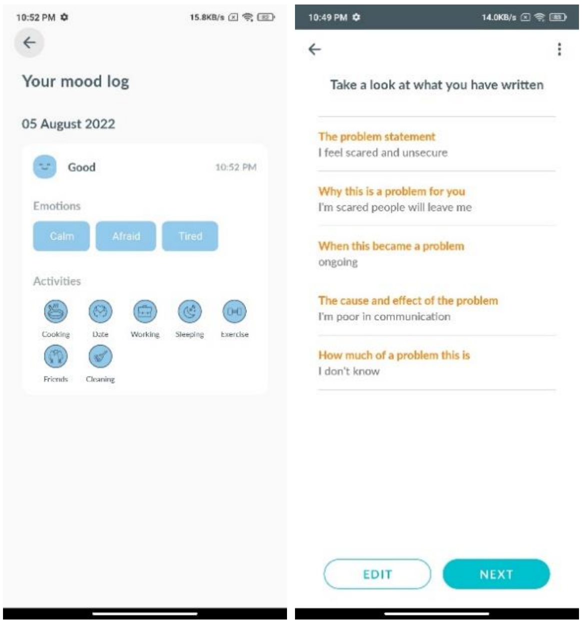


Figure 2.11 Homepage of Amaha Application

After completing the stress assessment, users will reach the application's homepage, as shown in Figure 2.11. Users can view the customized mental health therapy plan on the homepage, which lasts for a few weeks. After that, users must retake the assessment after completing weeks of online courses to re-evaluate the stress level. Besides that, users can also attend additional courses to deal with different negative feelings in daily life. In addition, the application encourages users to build emotional strength by practising relaxation and physical activities designed by experts and professional therapists. The activities can be added to the Amaha goal, as shown in figure 2.11, so users will get a daily reminder to complete the goal. The application also prepares a section for users to create their personal goals for greater user flexibility. This function lets users feel calm, maintain a journal, sleep better, and build lasting habits for good mental health [16].

The application claims to have accurate stress assessment results and a built-in scientific mood tracker. The therapy-based mood tracker can build self-awareness, care for users' moods, and produce more profound insights into mental health. The users can record their daily moods such as angry, sad, calm, and happy. In addition, users at Amaha can record both positive and negative feelings by writing down notes to describe their day. Lastly, the application will generate an insight consisting of a weekly mood chart and lifetime mood count, as given in Figure 2.12.



Your mood log

05 August 2022

Good 10:52 PM

Emotions: Calm, Afraid, Tired

Activities: Cooking, Date, Working, Sleeping, Exercise, Friends, Cleaning

Take a look at what you have written

The problem statement
I feel scared and unsecure

Why this is a problem for you
I'm scared people will leave me

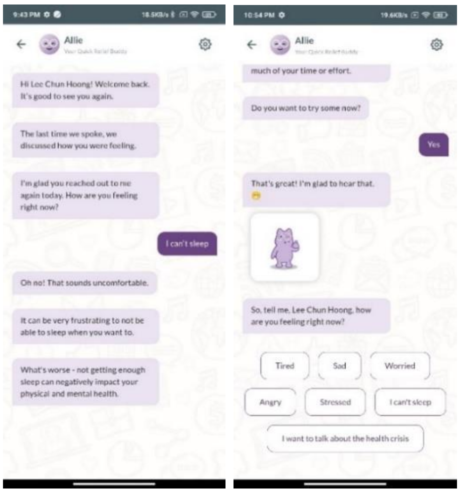
When this became a problem
ongoing

The cause and effect of the problem
I'm poor in communication

How much of a problem this is
I don't know

Figure 2.12 Mood log/tracker

The application also has one built-in chatbot, which assists users in overcoming stress, anxiety, or sleeping issues, as presented in Figure 2.13. The intelligent chatbot is called Allie. Amaha engineers and experienced therapists design it. During the communication with Allie, the chat data is encrypted and stored on encrypted servers where the data are not shareable with other third-party sources. Even though the chatbot can communicate with users with a preprogrammed algorithm, it can recommend any relevant relaxation activities to users to boost mental health. However, it is not encouraged to use the Allie chatbot for crisis assistance because users need to reach out to a crisis hotline if users are experiencing suicidal thoughts, self-harm or severe mental illness.



Hi Lee Chun Hoong! Welcome back. It's good to see you again.

The last time we spoke, we discussed how you were feeling.

I'm glad you reached out to me again today. How are you feeling right now?

I can't sleep

Oh no! That sounds uncomfortable.

It can be very frustrating to not be able to sleep when you want to.

What's worse - not getting enough sleep can negatively impact your physical and mental health.

much of your time or effort.

Do you want to try some now?

Yes

That's great! I'm glad to hear that. 😊

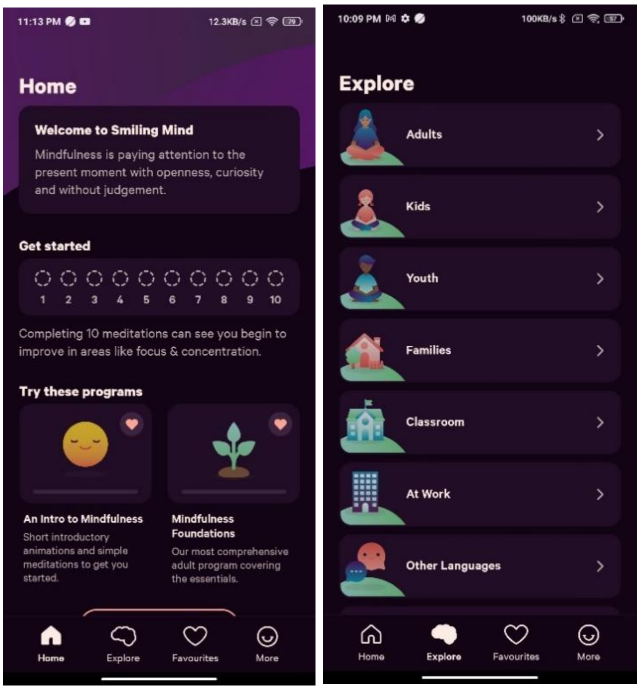
So, tell me, Lee Chun Hoong, how are you feeling right now?

Figure 2.13 Allie chatbot, which recommends relaxation activities to users

In conclusion, the application has a scientific stress assessment and mood tracker as the advantages of providing stress results in detail. Users can clearly understand their mental health status once the application tracks user problems. After that, they can set an Amaha goal, where the application customizes the mental health plan for users to carry out daily, or users can create a personal goal with any suitable physical and psychological activities set by the users. However, the disadvantage of this application is that users might find it hard to remove the registered programme if they would like to proceed with the newly created programme. Furthermore, users cannot see the previous additional notes during mood tracking as they only can check the mood counter as graphical information.

2.3.4 Smiling Mind

Smiling Mind is a mental health application designed for all ages because the programs in the application are designed for kids, youth, and adults [17]. It is a free mindfulness meditation application developed by psychologists and educators in Australia to help bring balance to the life of the users. The programs serve the purposes of dealing with pressure, stress, and daily life challenges. Mindfulness is always prioritized in Smiling Mind because it can create calm, regulate emotions, relax users, improve concentration, and increase productivity.



Home

Welcome to Smiling Mind

Mindfulness is paying attention to the present moment with openness, curiosity and without judgement.

Get started

Completing 10 meditations can see you begin to improve in areas like focus & concentration.

Try these programs

An Intro to Mindfulness

Short introductory animations and simple meditations to get you started.

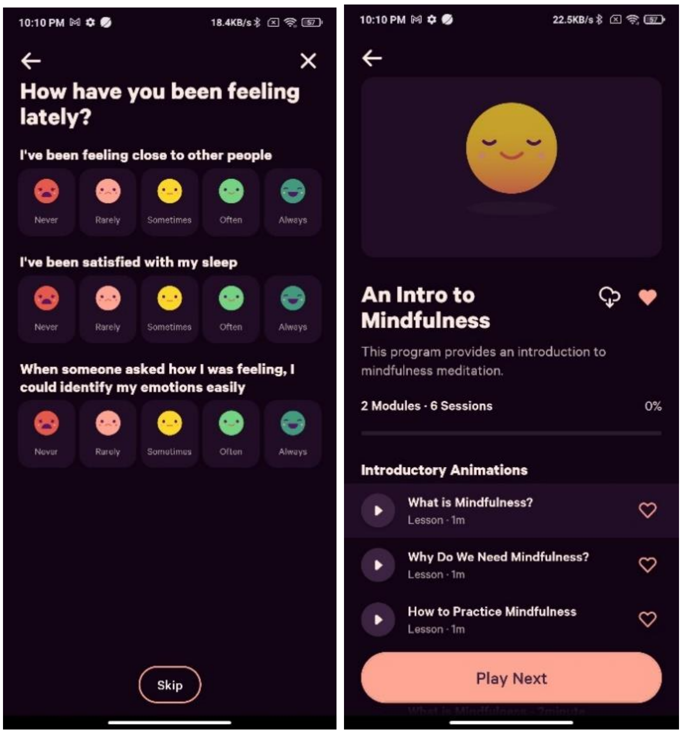
Mindfulness Foundations

Our most comprehensive adult program covering the essentials.

Explore

Figure 2.14 Homepage of Smiling Mind and program exploration

According to Figure 2.14, the users can quickly access their favourite saved meditation program. The application's design ensures a clean layout as simplicity is maintained. The meditation program library is filtered and categorized into different categories so users can search the program according to age, workplace, or language. For instance, the programs are designed to support adults to manage better the pressure and challenges of busy and modern lives. Besides that, the application has dedicated sleep programs for all ages to assist users in getting into sleep and improving overall sleep quality. Besides filtered by age, the programs are designed for places such as bedtime, workplaces, schools, and sporting clubs [18]. For example, the meditation programs in the classroom are structured programs designed for educators looking to bring mindfulness into their classrooms. The meditation programs are guided with smooth and calming voices and are included meditations in Indigenous Australian languages, Arabic, and Dari.



How have you been feeling lately?

I've been feeling close to other people

I've been satisfied with my sleep

When someone asked how I was feeling, I could identify my emotions easily

An Intro to Mindfulness

This program provides an introduction to mindfulness meditation.

2 Modules · 6 Sessions 0%

Introductory Animations

What is Mindfulness? Lesson · 1m

Why Do We Need Mindfulness? Lesson · 1m

How to Practice Mindfulness Lesson · 1m

Figure 2.15 Meditation assessment in the program

Before the users start their meditation class, they must answer the questions shown in Figure 2.15 above. Users can choose to answer or skip the assessment, but the answers before the meditation help users better define their problems so they can be tackled effectively. These questions are designed to show users how their mindfulness practice may be helping impact areas of their life. During the meditation session, users can choose to remove the background music. After each session, users can view the program’s progress under the progress bar so users can easily understand the progress daily or weekly. However, the application only focuses on providing a meditation program but not other activities such as breathing, relaxation, and physical activities [19].

In short, Smiling Mind is indeed an excellent application to relieve stress and regulate emotions because it provides different types of meditation programs, even for kids age. The programs are wide variety as they are designed anytime during the day. In addition, some of the meditation programs are recorded in different languages for the particular language-speaking community. On the contrary, the application has some disadvantages because it does not have a dedicated stress assessment to generate their mental health status accurately. The only way to measure their well-being is to complete the questionnaire before the meditation session.

2.4 Critical Analysis

This section discusses the critical analysis of the stress management applications, namely Headspace, Shine, Amaha, and Smiling Mind, which are reviewed in Section 2.3.

2.4.1 Headspace

To highlight, this app's benefit is that it offers a variety of mindfulness-based meditation exercises to assist various users in overcoming stress and anxiety. Users can also research mindfulness techniques from many angles, including breathing exercises, focus music, and sleep meditation. Besides that, the application will help users to start the day by recommending activities from morning to night. The drawback of this tool is that users can only do stress tests once a month. This is because users cannot remember every scenario from the previous month.

Table 2.1 Advantages and Disadvantages of Headspace

Advantages	Disadvantages
- Wide variety of meditation exercises with recommendation - Users can search for mindfulness tips if they have any problems	- A limited number of times for users to perform stress evaluation tests.

2.4.2 Shine

In a word, the application's benefit is that it offers several search possibilities, allowing users to locate the perfect meditation class for them. In addition, it provides the check-in function to allow users to express their gratitude to save as a record. The application also offers a discussion forum where users can exchange encouraging words. The application does not, however, have a component for assessing stress or negative thoughts. The check-in feature allows users to reflect on themselves; it cannot monitor feelings of rage, worry, or anxiety.

Table 2.2 Advantages and Disadvantages of Shine

Advantages	Disadvantages
- It offers many filtering and searching options in search engine - Check-in function to express gratitude as a positive thought - A discussion room is provided to allow users to post positive words	- Cannot monitor other negative thoughts from users

2.4.4 Amaha (InnerHour)

The application's advantages of giving detailed data for stress include a scientific stress evaluation and mood tracker. Once the application keeps track of user issues, users may quickly grasp their mental health state. After that, users can make a personal goal with any appropriate physical and psychological activities chosen by the users, or they can set an Amaha goal, where the application customizes the mental health plan for users to carry out daily. The drawback of this application is that users could find it challenging to unregister a programme if they want to move on to a newly generated programme. Additionally, users cannot view the earlier additional remarks while tracking their mood because they can only view the mood counter as graphical data.

Table 2.3 Advantages and Disadvantages of Amaha (InnerHour)

Advantages	Disadvantages
- Scientific stress evaluation and mood tracker - Choose to follow an in-app goal or set up a personal goal to achieve mental health	- Users cannot remove a registered programme as there is no cancellation option. - Previously made notes cannot be viewed in the mood tracker.

2.4.5 Smiling Mind

Smiling Mind offers a variety of meditation sessions, even for younger users, making it an excellent tool for reducing stress and controlling emotions. The programmes come in a wide range and can be created at any time of day. Additionally, some meditation programmes are recorded in several languages for the community that speaks those languages. On the other hand, the programme has several drawbacks because it lacks a dedicated stress evaluation to determine their mental health status accurately. Their well-being will be evaluated only by completing the questionnaire before the meditation session.

Table 2.4 Advantages and Disadvantages of Smiling Mind

Advantages	Disadvantages
- The application offers a variety of meditation sessions for different ages. - Some meditation programs are recorded in a different language	- It lacks a dedicated stress evaluation to determine their mental health status accurately.

CHAPTER 3: System Design

3.1 Overview

This chapter will explain the flow of the proposed application and each module in the project by illustrating different design diagrams such as the system architecture, use case diagram, and activity diagram. The use case description is a suitable way to outline the behavior of the system from the user's perspective when responding to a request. This too helps to explain well how users perform certain tasks in the proposed application.

3.2 System Architecture

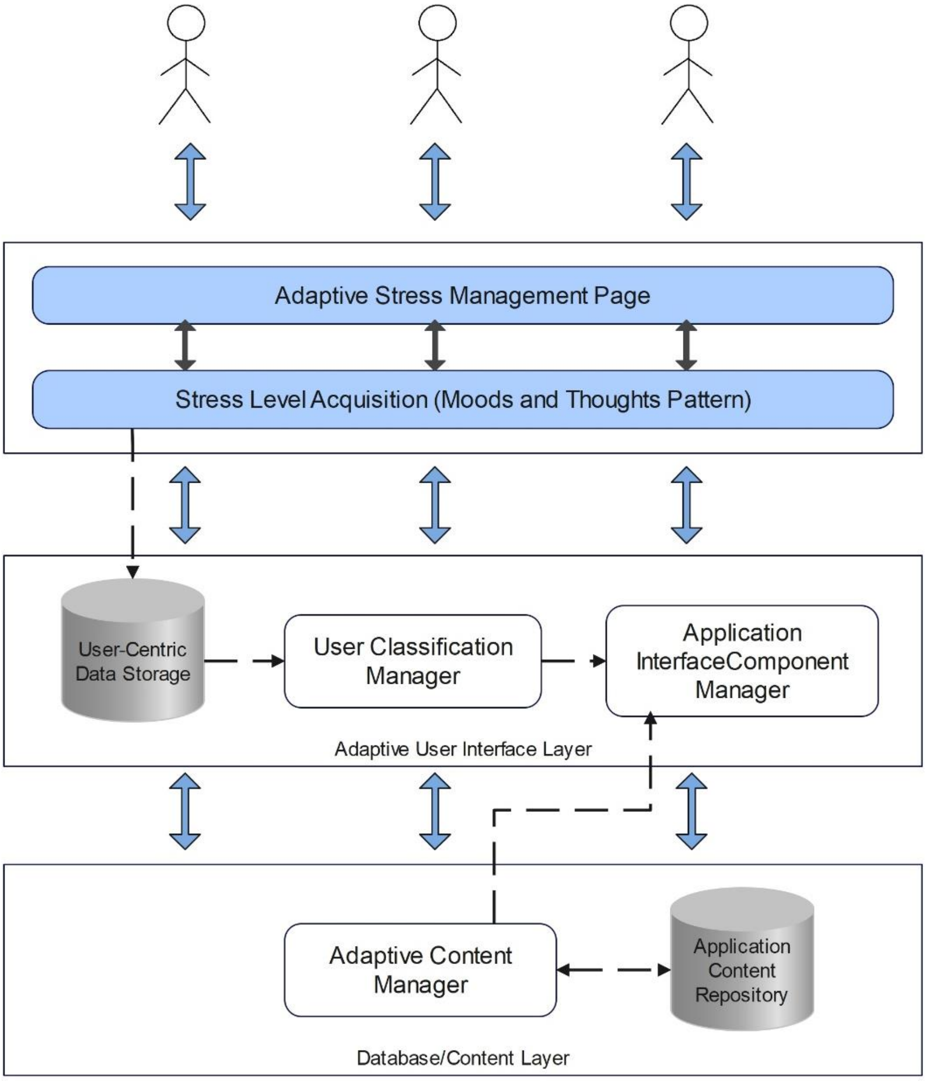

Adaptive Stress Management Page
Stress Level Acquisition (Moods and Thoughts Pattern)
User-Centric Data Storage
User Classification Manager
Application InterfaceComponent Manager
Adaptive User Interface Layer
Adaptive Content Manager
Application Content Repository
Database/Content Layer

Figure 3.1 System architecture of the proposed stress mental health application

The system architecture is divided into several layers. Beginning with the topmost layer, users will interact with the stress level management application. This layer provides users with access to most of the application's features. Notably, this includes the DASS-21 stress score questionnaire and [2] the MBSR program [8] aimed at reducing stress levels and monitoring stress over time. When new users join the application, their first step is to either log in or register a new account. This account linkage ensures that user program data is associated correctly for future reference. This front-end section is designed for user-friendly access to features through the user interface. Any subsequently created data flows down to the lower application logic layer and then into the database.

Within the application, mood and thought data collection is consistently conducted through various features. In the proposed application, the mood check-in function plays a vital role for all users, particularly those newly registered. It lets users understand their initial mental state before engaging in stress-relieving functions and programs. The DASS-21 questionnaire and mood check-in function serve as crucial components here, allowing users to track their mood and answer the questionnaire at specific intervals. Additionally, the mini-AI chatbot learns from user input messages when users communicate about their feelings, stress, or other topics. It generates relevant responses based on the input to address user stress.

Transitioning to the application logic layer, user-centric data storage resides below the previous layer, as the repository is linked to individual user accounts. Google Firebase is utilized to store user data in a database collection. Personal information for all users is registered and stored in this user database. Any data generated from the aforementioned application functionalities is stored within the user's account-specific collection. This approach avoids a shared database where user-related data might become mixed up. User classification management follows, where the application identifies the current user based on their registered user ID in Google Firebase. This allows for the connection to their own attribute data.

The application-interface component manager handles backend services within the application. It comprises self-contained modules and pieces of code, each performing specific functions. These components interact with the Firebase Database for data processing, user authentication, storage, and communication with external services. They are the portions of the application that users directly interact with, serving as the bridge between the graphical user interface (GUI) and application programming interface (API). This module facilitates the coordination and optimization of interactions between components and interfaces.

Lastly, the database and content layer form the bottommost part of the system architecture. The adaptive content manager dynamically adjusts and delivers content to users based on preferences, behaviours, and context. This ensures personalized and relevant content is provided in real-time. Moreover, it offers updates as user data and context change. This layer serves as an intermediary, linking both the database repository and component manager above. For instance, the users can receive different stress scores based on their responses to the DASS-21 questionnaire, spanning from normal to extremely severe. During this process, the system generates accurate diagnosis reports from the database according to the stress scores obtained. Thus, the system delivers tailored content to each user, presenting information most relevant to their needs.

3.3 Use Case Diagram

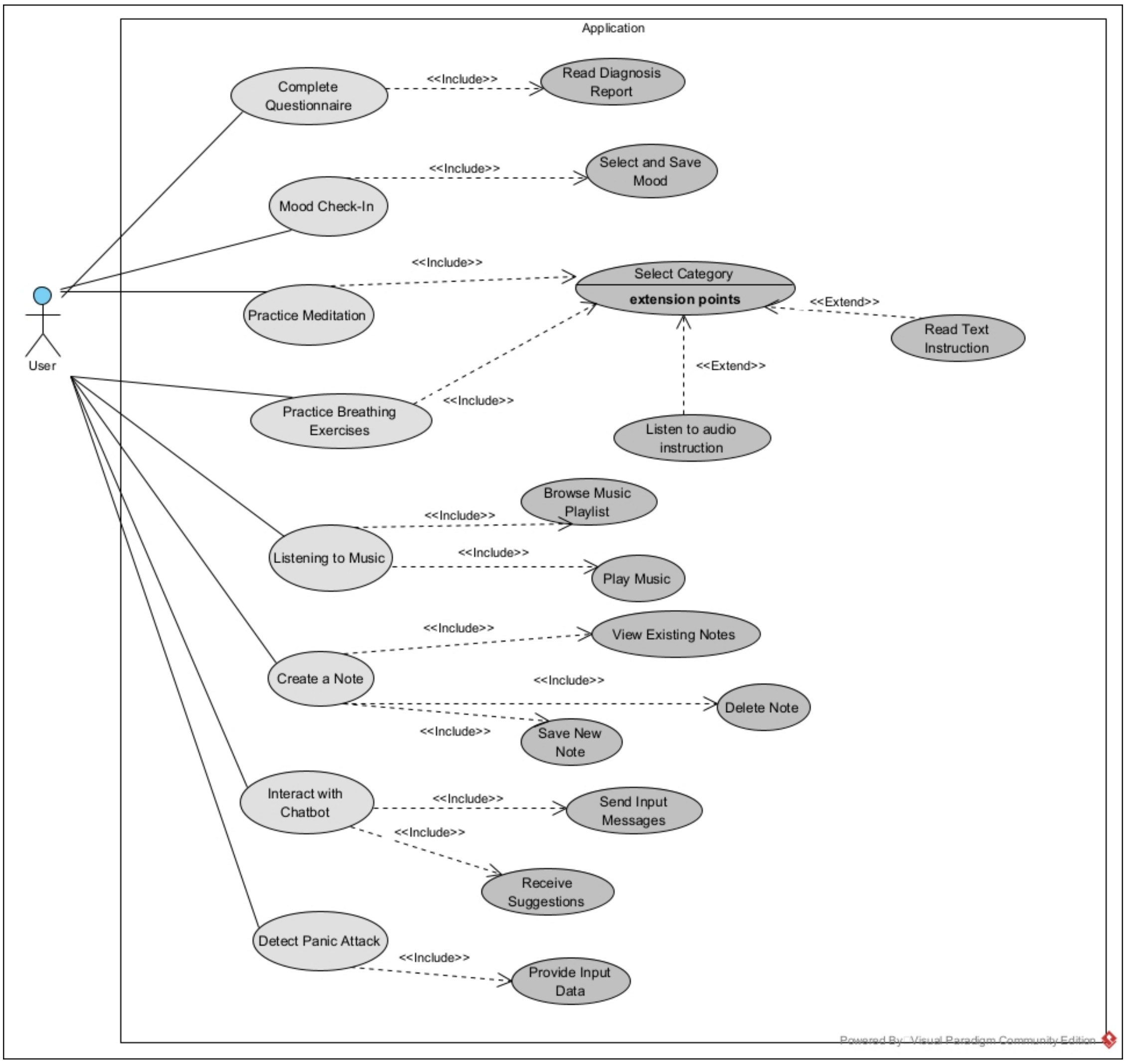


Figure 3.2 Use case diagram of the proposed stress mental health application

Figure 3.2 is a use case diagram that shows the tasks that users can perform on the application. The use case descriptions for the tasks are discussed in the following section, Section 3.4.

3.4 Use Case Description

Table 3.1 Use Case Description for “Complete Questionnaire” Use Case

Use Case ID	UC001	**Use Case Name**	Complete Questionnaire
Primary Actor	User		
Brief Description	Users are required to complete the DASS-21 questionnaire before accessing the application homepage. This step allows them to acquire a stress score value and gain a better understanding before utilizing the stress-reduction functions.		
Trigger	User selects the “Answer DASS-21 Questionnaire” option.		
Precondition	Users log into the application using their respective accounts.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User accesses the questionnaire page.	
	2	User starts answering the first question.	
	3	System advances to the subsequent question once the user selects an answer, continuing until the final question.	
	4	System calculates the stress score and upload to the database.	
	5	System retrieves and presents the diagnosis report from the database according to the stress score range obtained by the user.	
	6	User clicks the “Confirm” button and navigate to the application homepage.	
Sub Flow – Displaying the meaning of the question	2a.1	Users click the “Don’t understand the question?” button.	
	2a.2	System provides question explanations to clarify the meaning for users who may not understand the question's intent.	
	2a.3	User closes the dialog box by clicking the “OK” button.	

Table 3.2 Use Case Description for “Mood Check-In” Use Case

Use Case ID	UC002	**Use Case Name**	Mood Check-In
Primary Actor	User		
Brief Description	Users have the option to choose their overall mood throughout the day, and the selected mood will be uploaded to their user database. The mood options include neutral, happy, sad, angry, motivated, disappointed, and more.		
Trigger	User selects the “Check In” option.		
Precondition	User has accessed the application homepage.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User accesses the mood check-in page.	
	2	User selects the mood options provided in the list.	
	3	System provides a brief description of each mood beneath the list.	
	4	User clicks the “Confirm” button.	
	5	System receives the selected mood and uploads it into the user database, along with the current date.	
	6	System navigates the user back to the application homepage and updates the content of the homepage.	
	7	User views the mood history for the past 7 days, displaying the list of moods they have experienced during that period.	

Table 3.3 Use Case Description for “Practice Meditation” Use Case

Use Case ID	UC003	**Use Case Name**	Practice Meditation
Primary Actor	User		
Brief Description	Users can choose MBSR meditation exercises from a provided list categorized as Beginner, Intermediate, and Advanced. Upon selection, the system will present the chosen meditation exercise along with accompanying instructions.		
Trigger	Users can access the "MBSR" navigation option and select from any of the three category buttons: Beginner, Intermediate, or Advanced.		
Precondition	User has accessed the application homepage.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User accesses the MBSR activities page.	
	2	User selects the meditation categories among the three options.	
	3	System displays the meditation exercises available.	
	4	The user chooses a meditation exercise according to the selected category.	
	5	User clicks the “Start Now” button.	
	6	System presents textual instructions and offers audio guidance on the screen.	
	7	System performs Sub Flow based on the tasks performed by the user.	
Sub Flow – Reading Text Instruction	7a.1	User reads the text instruction of the meditation.	
	7a.2	Users can mark a symbol or icon beside each text instruction to indicate completion after successfully following the given instruction.	
	7a.3	Systems displays a dialog box indicating that the user has completed the meditation exercise.	
	7a.4	User clicks the “OK” button.	
Sub Flow – Listening to Audio Instruction	7b.1	User clicks the “Play” button.	
	7b.2	The system begins playing the audio, while also showing the meditation title and its duration.	
	7b.3	User clicks the “Stop” or “Pause” button after completion of the meditation exercise.	

Table 3.4 Use Case Description for "Practice Breathing Exercise" Use Case

<table>
<tr><td><b>Use Case ID</b></td><td>UC004</td><td><b>Use Case Name</b></td><td>Practice Breathing Exercise</td></tr>
<tr><td><b>Primary Actor</b></td><td colspan="3">User</td></tr>
<tr><td><b>Brief Description</b></td><td colspan="3">Users have the option to choose the MBSR Breathing Exercise from the provided selection of three exercises. This section offers three distinct breathing techniques, each accompanied by the exercise's name and duration. Users can engage in practicing the breathing exercises to alleviate stress and maintain a sense of calmness.</td></tr>
<tr><td><b>Trigger</b></td><td colspan="3">Users navigates to the "MBSR" section and scroll through the available breathing exercises.</td></tr>
<tr><td><b>Precondition</b></td><td colspan="3">User has accessed the application homepage.</td></tr>
<tr><td><b>Scenario Name</b></td><td><b>Step</b></td><td colspan="2"><b>Action</b></td></tr>
<tr><td><b>Main Flow</b></td><td>1</td><td colspan="2">User accesses the MBSR activities page.</td></tr>
<tr><td></td><td>2</td><td colspan="2">System displays the breathing exercises available.</td></tr>
<tr><td></td><td>3</td><td colspan="2">User clicks the "Start Now" button.</td></tr>
<tr><td></td><td>4</td><td colspan="2">System presents textual instructions and offers audio guidance on the screen.</td></tr>
<tr><td></td><td>5</td><td colspan="2">System performs Sub Flow based on the tasks performed by the user.</td></tr>
<tr><td><b>Sub Flow – Reading Text Instruction</b></td><td>5a.1</td><td colspan="2">User reads the text instruction of the breathing technique.</td></tr>
<tr><td></td><td>5a.2</td><td colspan="2">Users can mark a symbol or icon beside each text instruction to indicate completion after successfully following the given instruction.</td></tr>
<tr><td></td><td>5a.3</td><td colspan="2">Systems displays a dialog box indicating that the user has completed the breathing exercise.</td></tr>
<tr><td></td><td>5a.4</td><td colspan="2">User clicks the "OK" button.</td></tr>
<tr><td><b>Sub Flow – Listening to Audio Instruction</b></td><td>5b.1</td><td colspan="2">User clicks the "Play" button.</td></tr>
<tr><td></td><td>5b.2</td><td colspan="2">The system begins playing the audio, while also showing the breathing technique title and its duration.</td></tr>
<tr><td></td><td>5b.3</td><td colspan="2">User clicks the "Stop" or "Pause" button after completion of the breathing exercise.</td></tr>
</table>

Table 3.5 Use Case Description for “Listening to Music” Use Case

Use Case ID	UC005	**Use Case Name**	Listening to Music
Primary Actor	User		
Brief Description	Users can choose from a selection of calming music within the "MBSR Music" section. The page offers various music options categorized by weather or scenery, such as rain, ocean, forest, fire, and more.		
Trigger	User clicks on the “Calming Music” button.		
Precondition	User has accessed the application homepage.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User accesses the MBSR Calming Music Page.	
	2	System displays the music playlist.	
	3	User selects one of the music from the music playlist.	
	4	User clicks the “Play Now” button.	
	5	System starts playing the selected music.	
	6	User clicks on the displayed music name.	
	7	System presents a comprehensive music player with full controls for users to manage the selected music.	

Table 3.6 Use Case Description for "Create a Note" Use Case

Use Case ID	UC006	**Use Case Name**	Create a Note
Primary Actor	User		
Brief Description	Users can access the "Notes" section to view their previously created notes. The existing notes will be presented in a horizontal list, including the note title and the timestamp of creation. Additionally, users can create new notes to document their emotions or daily experiences.		
Trigger	Users clicks on the "Note" button.		
Precondition	User has accessed the application homepage.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User accesses the "Notes" page.	
	2	System showcases a list of previously created notes, fetched from the database.	
	3	User clicks the "Add" button.	
	4	System provides the "Create Note" page, enabling users to draft a new note.	
	5	User enters a title for the new note and proceeds to type the content of the note's narrative.	
	6	User clicks the "Save Note" button.	
	7	System saves the newly created note along with the current timestamp to the database, then redirects the user back to the "Notes" page.	
	8	System automatically refreshes the page and presents the most recent list of notes on the screen.	
Sub Flow – Edit Existing Note	2a.1	User selects one of the existing notes from the list.	
	2a.2	System retrieves data from the database and displays the content of the notes on the screen.	
	2a.3	User edits either the note title or the note content.	
	2a.4	User clicks the "OK" button.	
	2a.5	User clicks the "Save Note" button.	
	2a.6	System saves the newly created note along with the current timestamp to the database, then redirects the user back to the "Notes" page.	
	2a.7	System automatically refreshes the page and presents the most recent list of notes on the screen.	

Sub Flow – Delete Note	2b.1	User clicks the “Rubbish bin” icon under the note.
	2b.2	System displays a dialog box to confirm the user's intention to delete the note.
	2b.3	User clicks the “Yes” button.
	2b.4	System automatically refreshes the page and presents the most recent list of notes on the screen.

Table 3.7 Use Case Description for "Interact with Chatbot" Use Case

Use Case ID	UC007	**Use Case Name**	Interact with Chatbot
Primary Actor	User		
Brief Description	Users can initiate communication with the mini-AI chatbot application. Through interactions with the chatbot, users can discuss their stress levels, mental condition, or recently obtained stress score value, as these serve as input messages. In response, the chatbot offers relevant suggestions and advice to help users monitor and alleviate stress.		
Trigger	User clicks on the "Chatbot" section on "Profile" page.		
Precondition	User has accessed the application homepage.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User accesses the "Profile" section.	
	2	User clicks on "Try HepiPal" button.	
	3	System presents a welcoming message from the chatbot as the default greeting to the user.	
	4	User sends input messages.	
	5	System forwards the input message to the backend API service for processing.	
	6	System provides the user with the output message in response.	
	7	User receives the responded message from the system and subsequently send a message back.	

Table 3.8 Use Case Description for "Detect Panic Attack" Use Case

Use Case ID	UC008	**Use Case Name**	Detect Panic Attack
Primary Actor	User		
Brief Description	Users can engage with the feature powered by the machine learning KNN algorithm, designed to predict future panic attacks. By supplying the necessary input, particularly academic background details, the trained model will classify the output to determine whether the user is likely to experience a panic attack in the future.		
Trigger	User clicks on the "Predict Panic Attack" section on "Profile" page.		
Precondition	User has accessed the application homepage.		
Scenario Name	**Step**	**Action**	
Main Flow	1	User accesses the "Profile" section.	
	2	User clicks on "Predict Panic Attack" button.	
	3	System presents a form for users to input the necessary information required for classification.	
	4	Users enter the input data, including age, gender, course of study, current academic year, and other relevant information.	
	5	System passes the collected input data to the trained KNN model to generate an outcome.	
	6	System showcases the prediction result to the user on the screen.	

3.5 Activity Diagram

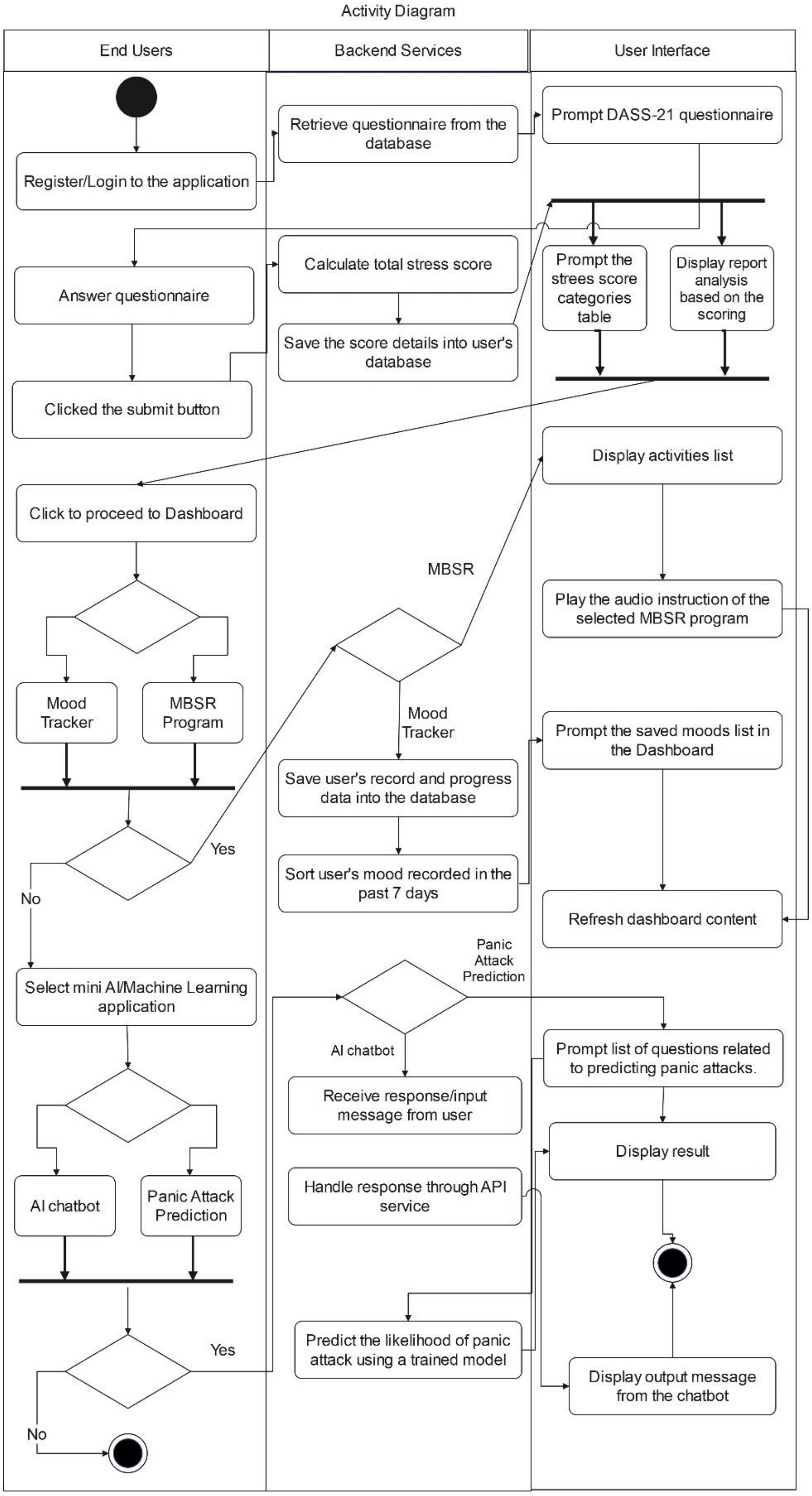

Activity Diagram
End Users
Backend Services
User Interface
Register/Login to the application
Retrieve questionnaire from the database
Prompt DASS-21 questionnaire
Answer questionnaire
Calculate total stress score
Prompt the strees score categories table
Display report analysis based on the scoring
Save the score details into user's database
Clicked the submit button
Display activities list
Click to proceed to Dashboard
MBSR
Play the audio instruction of the selected MBSR program
Mood Tracker
MBSR Program
Mood Tracker
Prompt the saved moods list in the Dashboard
Save user's record and progress data into the database
Yes
No
Sort user's mood recorded in the past 7 days
Refresh dashboard content
Panic Attack Prediction
Select mini AI/Machine Learning application
AI chatbot
Prompt list of questions related to predicting panic attacks.
Receive response/input message from user
Display result
AI chatbot
Panic Attack Prediction
Handle response through API service
Yes
Predict the likelihood of panic attack using a trained model
Display output message from the chatbot
No

Figure 3.3 Activity diagram for the proposed mobile application

One of the application's objectives is to provide a user-friendly user interface and various valuable programs and activities to reduce user stress. The users must take part in the stress score evaluation at least once before they access the application features. The application will ensure the users are fully aware of their mental conditions to carry out a more effective activities to maintain their mental health. Therefore, a series of procedures are prepared during the development. Figure 3.3 is an activity diagram involving the end users, backend services, and user interface, which describes the application flow. When a new user completes the user profile registration, the system will prompt the DASS-21 questionnaire on the screen to let the user evaluate their stress score. When the users answer all seven questions as listed in Appendix 1, a button is provided. When clicked, the system will prompt an explanation of the question to assist the users for better understanding before answering. After the users submit the score answers, the backend services will calculate the total stress score and save the information into the user database. Based on the user's response, a report with deeper analysis will be retrieved from the database and shown to the users, explaining the cause of the stress level and a simple recommendation to reduce the stress score.

Users are expected to better understand their stress issues from the report provided. After that, they can proceed to the dashboard to access all the application features. For example, one of the essential features of the application is the mood tracker, which allows the users to record their moods of the day for the past seven days based on the options provided in the list. Different descriptions will be shown under the list when the users click on the moods. When users submit the mood of the day, the system will save the mood and the current timestamp in the database.

Moreover, users can select to carry out MBSR programs, such as the Quick Destress section, whenever they feel sudden stress and need a quick solution to remain calm and relaxed. The system will then prompt a list of the Quick Destress options to the users. These short activities mostly take less than 5 minutes, so users can listen to the audio instruction and keep track of their activity progress.

Subsequently, users will have the option to utilize the integrated mini-AI/machine learning features. One of the features provide a mini-AI chatbot, as well as functionality for panic attack prediction and classification. For instance, if users select the AI chatbot as a platform for communication or to express themselves, where this feature will be driven by an online API service. The chatbot will receive input messages from users, which will then be forwarded to the API service to generate corresponding output messages. In essence, it operates similarly to a messaging application where users provide inputs while the chatbot consistently furnishes relevant responses. Consequently, users can engage with the chatbot whenever they wish to share their emotions or seek suggestions to address stress.

Simultaneously, users can choose to access the panic attack prediction function, developed specifically for students. The function will display certain questions to users to capture their current educational background info. Later, the integrated trained model within this function will process the provided inputs to classify whether users are at risk of experiencing a panic attack in the future. By interacting with these machine learning-powered functions, users can obtain deeper insights into their mental health condition, particularly in relation to stress levels.

CHAPTER 4: System Methodology/Approach

4.1 Proposed Method/Approach

Traditional software engineering consists of formalized quality, assurance, analysis, and design methods supported by Computer Aided Software Engineering (CASE) tools and controlled and rigorous software development processes. This was viewed as the best way to achieve better software other than careful project planning. Besides that, this software engineering method mainly concentrated on extensive system development. However, using the traditional waterfall method in small and medium-sized systems could lead to a significant overhead that dominates the software development process [20]. Therefore, in order to develop the proposed stress management mobile application, agile methodology has been chosen. In agile methodology, program specification, design, and implementation are inter-leaved and promise frequent delivery of the latest versions for evaluation [20].

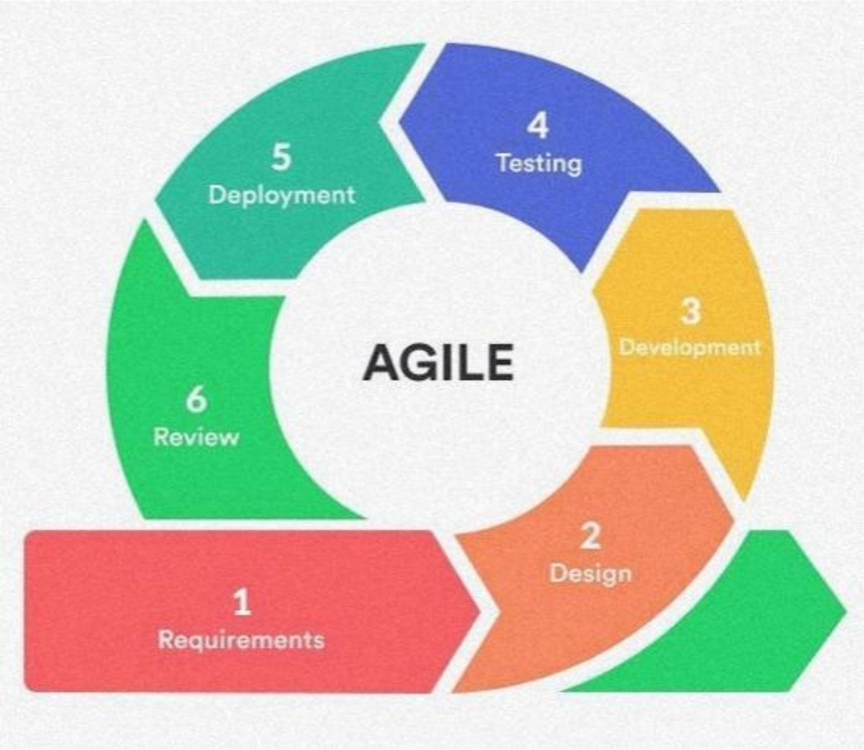


Figure 4.1 Agile Methodology in System Development [20]

Due to its adaptability, early delivery, and flexible life cycle, the agile technique, as shown in Figure 4.1, is ideally suited for rapid and efficient software development. Implementing agile project management approaches contributes to the development process's improvement through increased and effective communication, higher quality, better risk analysis, and less overspending [21]. Besides that, agile methodology is best used in a small team with base agile methodology knowledge and skills [20]. Using agile methodology, development teams should always be ready to embrace changes and accept that requirements will evolve throughout the project. In this proposed stress management application, agile methodology can adapt to changes at any time because some new features might be considered for implementation inside the application.

The development of this stress management application is best suited to the agile technique. This methodology ensures each phase's quality by dividing the application development life cycle into smaller modules, including analysis, design, programming, development, and project management. Additionally, each phase is supported by integrated documentation and quality testing. This model maximizes resource usage by dividing requirements into smaller, more manageable units.

According to [21], The agile Software Development Life Cycle (SDLC) approach prioritizes adapting processes and customer satisfaction by quickly providing a functional software solution. Agile methodologies divide the product into several iterations of smaller incremental builds, including planning, analyzing, architectural design, coding, unit testing, acceptance testing, delivery, and feedback. Software development teams can prototype the anticipated user interface at the same time. For example, during the design phase, the team examines how functions should be introduced and names the essential resources, including the programming language, syntactic libraries, and fundamental frameworks. Then, during the development phase, the product is delivered in separate sprints, each designed to improve the current version of the product. The initial release will likely undergo many changes to improve functionality and new features. As a result, using agile methodology in this application results in higher-quality functional designs being created.

4.2 System Requirement

4.2.1 Hardware Requirements

Table 4.1 Laptop specification

Hardware component	**Specification**
Processor	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz
Memory (RAM)	32 GB RAM
Disk space	480 GB SSD
Display	15-inch Laptop Screen

Table 4.2 Smartphone specification

Hardware component	**Specification**
Display resolution	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz
Operating system	Android 13, MIUI Global 14.0.4
RAM	8 GB
Manufacturer	Xiaomi

4.2.2 Software Requirements

Table 4.3 Software specification

Hardware component	**Specification**
Operating system	Windows 11 Home (64-bit)
Development Tool	Visual Studio Code React Native Expo Python
Development Platform	Google Firebase

4.2.2.1 Development Tool

4.2.2.1.1 Visual Studio Code

Visual Studio Code is a source code editor that supports languages like Java, Python, Javascript, C++, HTML, etc. It is lightweight and can run on Mac OS X, Windows, and Linux.

4.2.2.1.2 React Native

With React Native, programmers can use the same framework to create mobile apps for both iOS and Android. Using React and Javascript to build the application is convenient for the developers.

4.2.2.1.3 Expo

Expo is a React Native-based tool to help developers to build and test mobile applications. It offers services to create, build and publish apps in the development environment, with pre-defined UI components and other resources. Using Expo to build apps instead of other simulators saves developers more computing resources.

4.2.2.1.4 Python

Python has become the de facto choice for developing machine learning models using various algorithms. Machine learning offers numerous practical applications within projects. Python is gaining its popularity even until now is mainly because of the ease of understanding and implementation. Extensive collection of libraries is available to facilitate the construction of machine learning models, including scikit-learn, pandas, numpy, and TensorFlow.

4.2.2.2 Development Platform

4.2.2.2.1 Google Firebase

Google Firebase offers various backend services for the developers, such as real-time database, authentication, storage, web hosting, and cloud messaging. In this project, authentication, and Firebase Firestore database store user information, especially the stress evaluation score and report analysis. It provides a console to monitor projects and the usage of the database of reads and writes.

4.3 Timeline

4.3.1 – Timeline of the FYP1

Weeks Progress	1	2	3	4	5	6	7	8	9	10	11	12	13
Complete Figma prototype and interface design	🟧	🟧											
Set up Google Firebase Firestore database		🟧											
Set up React Native environment and connect to Expo			🟧										
Insert MBSR program details and DASS-21 reports into the database				🟧									
Program user authentication between the application and Firebase					🟧	🟧							
Program the DASS-21 questionnaire interface							🟧						
Program the logic behind DASS-21 answer steps and report retrieval								🟧					
Program mood tracker feature									🟧	🟧			
Develop MBSR Quick Destress functionality										🟧	🟧		
System testing											🟧		
Write report											🟧	🟧	
Complete and submit the final report												🟧	
Presentation													🟧

Figure 4.2 Timeline of the FYP1

Referring to Figure 4.3, the first activity planned in the FYP1 timeline is to design the interface and prototype of the application in Figma. The next activity is setting up Google Firebase Firestore for data storage and retrieval. Besides that, the React Native development environment is created, and the Expo connection is established. After the Firestore database is set up, data from DASS-21 reports and MBSR program details are inserted before the program authentication of Firebase in the application is developed. Once the preliminary work of the database and development platform is set up, the development of the main functionalities of the application can be started. It is expected to take two weeks to develop the DASS-21 questionnaire interface and report generation based on the user's stress score. Later, the programming of the mood tracker and MBSR Quick Destress can proceed. This is followed by the FYP1 report write-up being conducted in parallel with the system validation. Finally, the proposed application's presentation and demonstration are conducted on Week 13.

4.3.2 – Timeline of the FYP2

Weeks Progress	1	2	3	4	5	6	7	8	9	10	11	12	13
Program note-taking feature	🟧	🟧											
Program the logic layer of the complete MBSR feature		🟧											
Complete the relaxing music interface			🟧	🟧									
Refactor code that program in FYP1					🟧								
Study related API services for AI chatbot						🟧							
Program the AI chatbot feature							🟧	🟧					
Program a KNN machine learning training model								🟧	🟧				
Finetuning model and evaluate the model performance										🟧			
Finalize all functionalities in the application											🟧		
Deploy application and system testing											🟧		
Write report												🟧	🟧
Presentation													🟧

Figure 4.3 Timeline of the FYP2

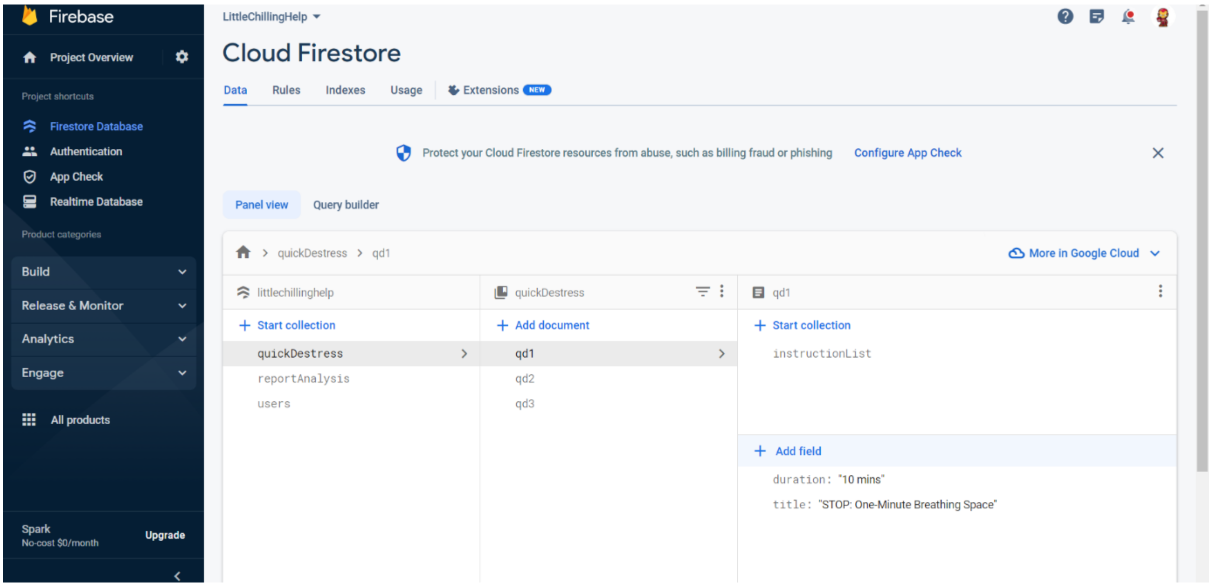
The main tasks to be completed in the starting timeline of FYP2 as shown in Figure 4.4 include the development of the note-taking feature, MBSR program logic layer, and refactoring code. Subsequently, a one-week period is anticipated for grasping the application of API services in order to develop an AI chatbot for the proposed application. Following this, a two-week interval will be dedicated to integrating the AI chatbot function from these services and concurrently designing the user interface. Moving on to Week 9, an additional AI model will be formulated utilizing the K-Nearest Neighbours (KNN) machine learning algorithm. During this phase, the model will undergo training and optimization of its hyperparameters, followed by a comprehensive performance evaluation. During Week 11, all functions in the application are expected to be finalized. Later, the application can be deployed for system testing. The FYP2 report write-up follows this. Finally, the FYP2 presentation and demonstration will be conducted in Week 13.

CHAPTER 5: System Implementation

5.1 Integrating Google Firebase Firestore Database

Google Firebase is a product developed by Google and offers various services to allow the developers to build and manage their apps with ease [22]. It is convenient to use as it requires no programming on the firebase backend infrastructure which makes the development faster and efficient. The services that will be used from the Firebase is the user authentication in application and Firestore database which provides cloud storage. NoSQL is used for the database to store and update the data.

Before a new React Native project is initiated, a storage medium needs to be identified for data storage including users and app data. Therefore, Google Firebase is chosen because of the efficiency is integrating NoSQL database and it comes with almost no cost. The Firebase plan that is currently being used is the Spark plan, which is free to use except it limits 50K times of document read and write per day. It the number goes beyond 50K, it will charge the total cost in the bill.



Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

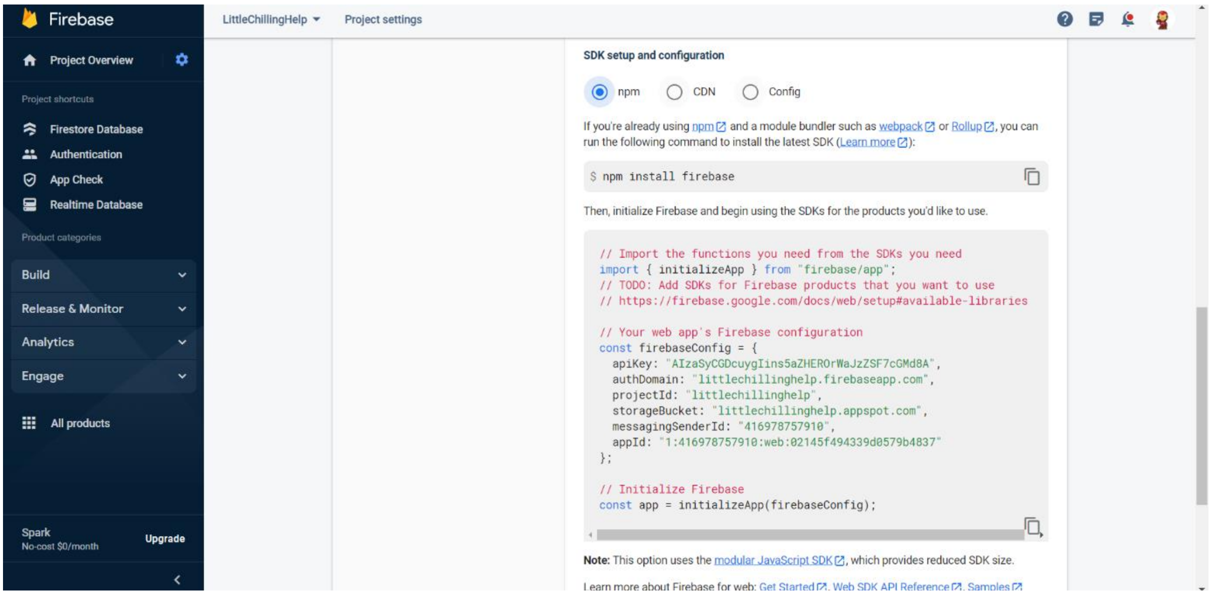
quickDestress > qd1

littlechillinghelp	quickDestress	qd1
quickDestress	qd1	instructionList
reportAnalysis	qd2	
users	qd3	

duration: "10 mins"
title: "STOP: One-Minute Breathing Space"

Figure 5.1 Overview of the Firebase Firestore database

Once the database is ready to use, a new project is initiated in React Native and Expo. The project is created with application name in Visual Studio Code and I will mostly code my file in a Javascript file, which is commonly used in React Native instead of ".tsx" file. Since the Firestore database is used but not the real-time database, some variables and parameters used need to be changed during copying the "config.js" file into the React Native. Figure 5.2 shows the config file in the Firebase setting and my React Native Project.



SDK setup and configuration

● npm ○ CDN ○ Config

If you're already using npm and a module bundler such as webpack or Rollup, you can run the following command to install the latest SDK (Learn more):

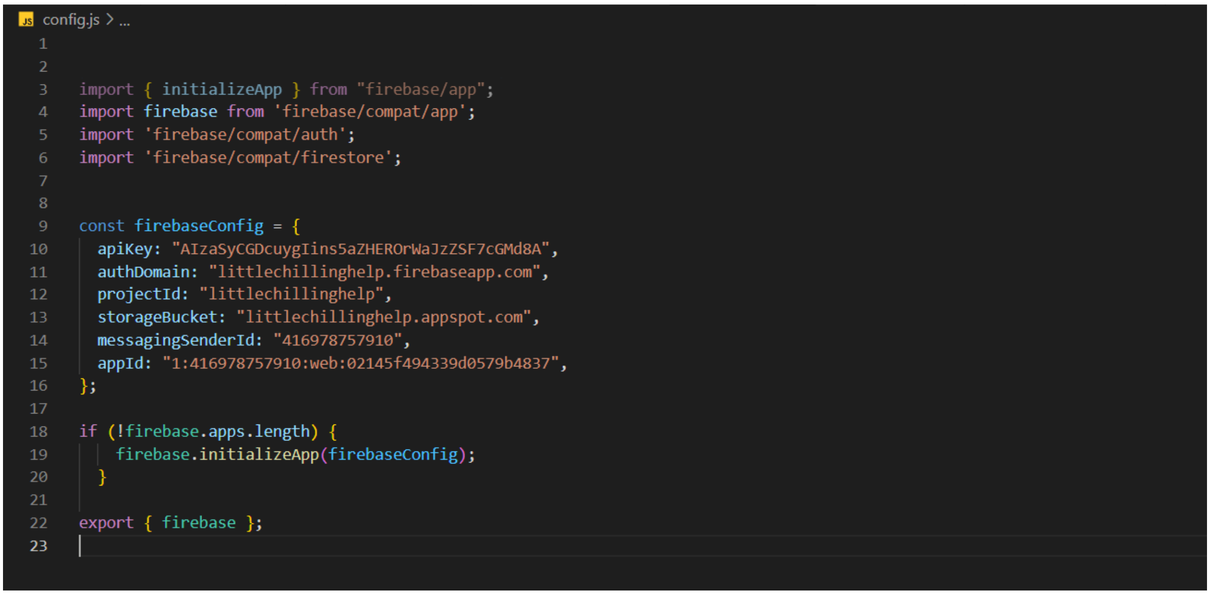
```
$ npm install firebase
```

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyCGDcuygIins5aZHEROrWaJzZSF7cGMd8A",
  authDomain: "littlechillinghelp.firebaseapp.com",
  projectId: "littlechillinghelp",
  storageBucket: "littlechillinghelp.appspot.com",
  messagingSenderId: "416978757910",
  appId: "1:416978757910:web:02145f494339d0579b4837"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Note: This option uses the modular JavaScript SDK, which provides reduced SDK size.

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

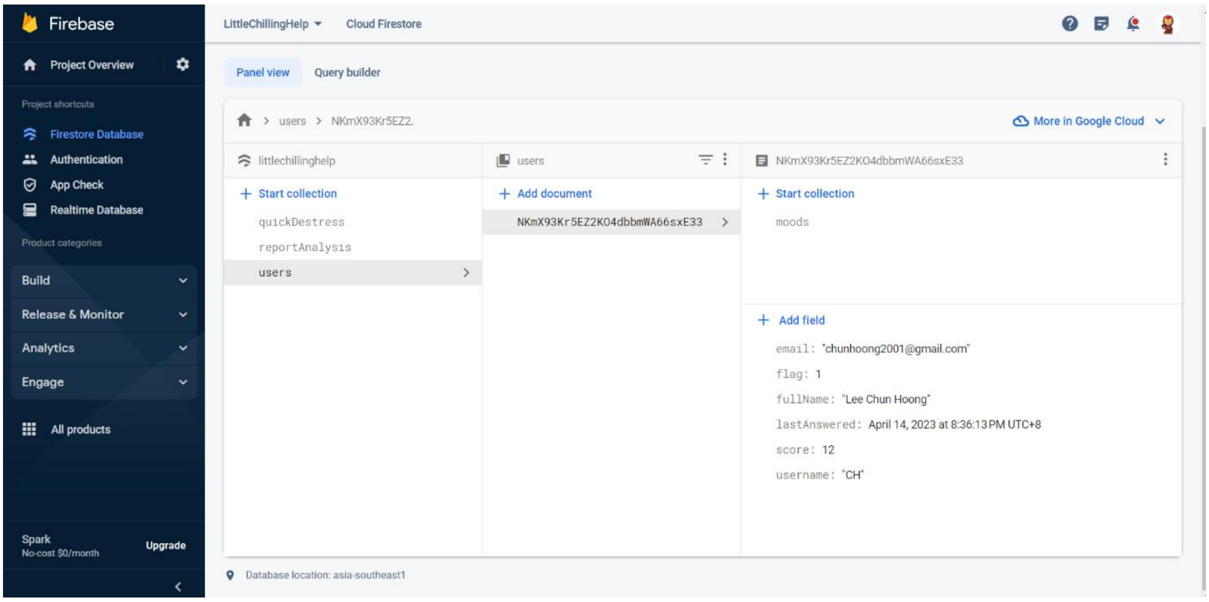


```
1
2
3   import { initializeApp } from "firebase/app";
4   import firebase from 'firebase/compat/app';
5   import 'firebase/compat/auth';
6   import 'firebase/compat/firestore';
7
8
9   const firebaseConfig = {
10    apiKey: "AIzaSyCGDcuygIins5aZHEROrWaJzZSF7cGMd8A",
11    authDomain: "littlechillinghelp.firebaseapp.com",
12    projectId: "littlechillinghelp",
13    storageBucket: "littlechillinghelp.appspot.com",
14    messagingSenderId: "416978757910",
15    appId: "1:416978757910:web:02145f494339d0579b4837",
16  };
17
18  if (!firebase.apps.length) {
19      firebase.initializeApp(firebaseConfig);
20    }
21
22  export { firebase };
23
```

Figure 5.2 Firebase database configuration in React Native

The recent version updates of the Firebase cause the import libraries to change. In Visual Studio Code, Firebase is imported from “compat/auth” and “compat/firestore” to apply the authentication, and data storage and retrieval module. Besides that, precaution steps were added in the configuration file, where line 18 to 20 checked the array length of the Firebase app. If the number of array length is greater than zero, then the Firebase will initiate a new app from the “firebaseConfig” provided. This step prevents multiple initiations of application in the first phase, as Firebase app is only initialized once.

Based on the Figure 5.1, most of the app data were inserted into the database by including the DASS-21 questionnaire report analysis and MBSR program details. Each entity is placed in different collections in the database including the user's database. For example, every user's data is stored in the "users" collection and a unique document ID to identify the users will be auto generated. Therefore, the need to create an extra new field to specify the User ID is no longer needed.



quickDestress
reportAnalysis
users

NKmX93Kr5EZ2KO4dbbmWA66sxE33

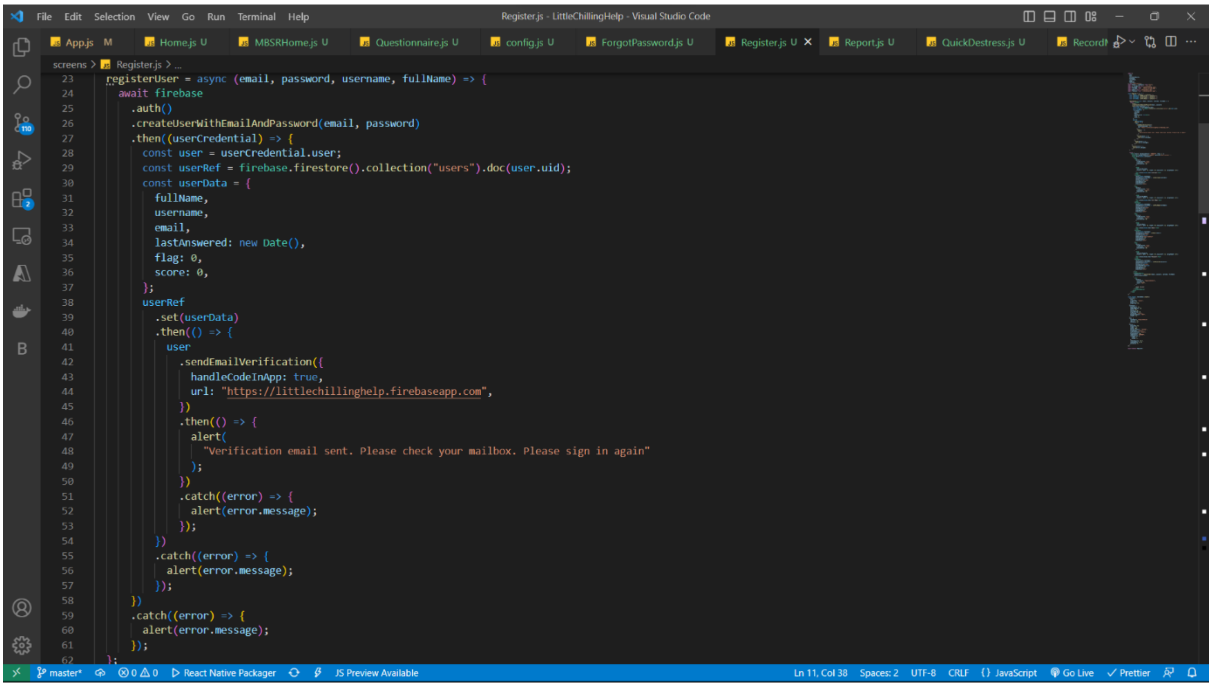
moods

email: "chunhoong2001@gmail.com"
flag: 1
fullName: "Lee Chun Hoong"
lastAnswered: April 14, 2023 at 8:36:13 PM UTC+8
score: 12
username: "CH"

Figure 5.3 Users collection in Firebase database

Furthermore, applied one of the functions is applied in Firebase, which is the user authentication to register and login purpose. Firebase provides a channel for the developers to allow users to sign in or create a new account easily without having the developers to hard code the program. In this project, email and password providers are used to let users to register an account. the user interface is programmed in React Native which will asked for users' information such as full name, username, email, and password. This approach requires a valid user email so an activation link will be sent through mail to the users to verify themselves before they can start using the application.

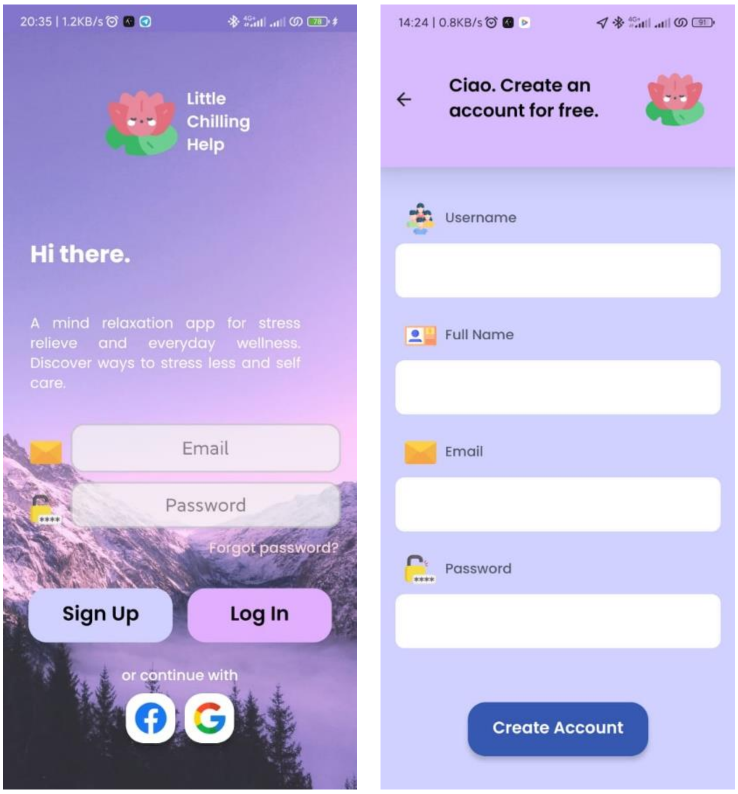
The code snippet shown below in Figure 5.4 is to identify the variables set by the users in the "<TextInput>" area and saves them into the user database. Besides the four variables mentioned above, other variables such as "lastAnswered", "score", and "flag" will be declared into the database with default value, since the users never answer the DASS-21 questionnaire if they are the new users.



File Edit Selection View Go Run Terminal Help
Register.js - LittleChillingHelp - Visual Studio Code
App.js M
Home.js U
MBSRHome.js U
Questionnaire.js U
config.js U
ForgotPassword.js U
Register.js U
Report.js U
QuickDestress.js U
screens > Register.js > ...
master*
React Native Packager
JS Preview Available
Ln 11, Col 38 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

```
  registerUser = async (email, password, username, fullName) => {
    await firebase
      .auth()
      .createUserWithEmailAndPassword(email, password)
      .then((userCredential) => {
        const user = userCredential.user;
        const userRef = firebase.firestore().collection("users").doc(user.uid);
        const userData = {
          fullName,
          username,
          email,
          lastAnswered: new Date(),
          flag: 0,
          score: 0,
        };
        userRef
          .set(userData)
          .then(() => {
            user
              .sendEmailVerification({
                handleCodeInApp: true,
                url: "https://littlechillinghelp.firebaseapp.com",
              })
              .then(() => {
                alert(
                  "Verification email sent. Please check your mailbox. Please sign in again"
                );
              })
              .catch((error) => {
                alert(error.message);
              });
          })
          .catch((error) => {
            alert(error.message);
          });
      })
      .catch((error) => {
        alert(error.message);
      });
  };
```

Figure 5.4 Code snippet of uploading the user's information to the database

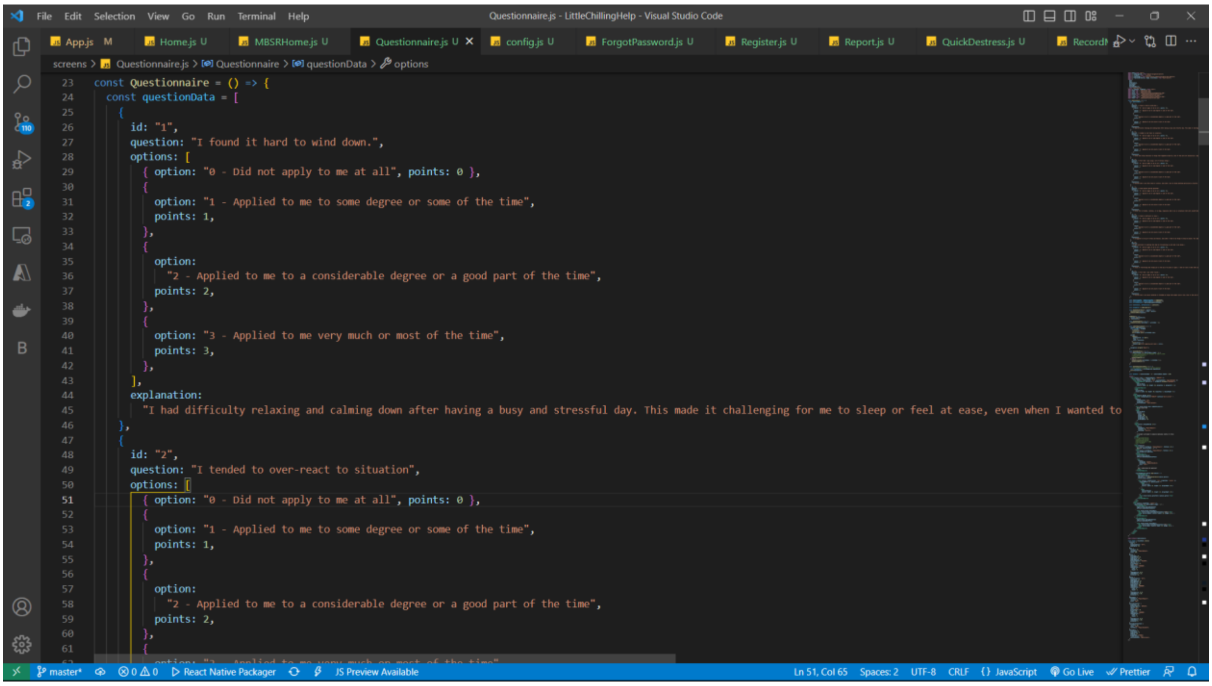


Little Chilling Help
Hi there.
A mind relaxation app for stress relieve and everyday wellness. Discover ways to stress less and self care.
Email
Password
Forgot password?
Sign Up
Log In
or continue with
Ciao. Create an account for free.
Username
Full Name
Email
Password
Create Account

Figure 5.5 Login Page (Left) and Register Page (Right) of the application

5.2 Developing the DASS-21 questionnaire interface

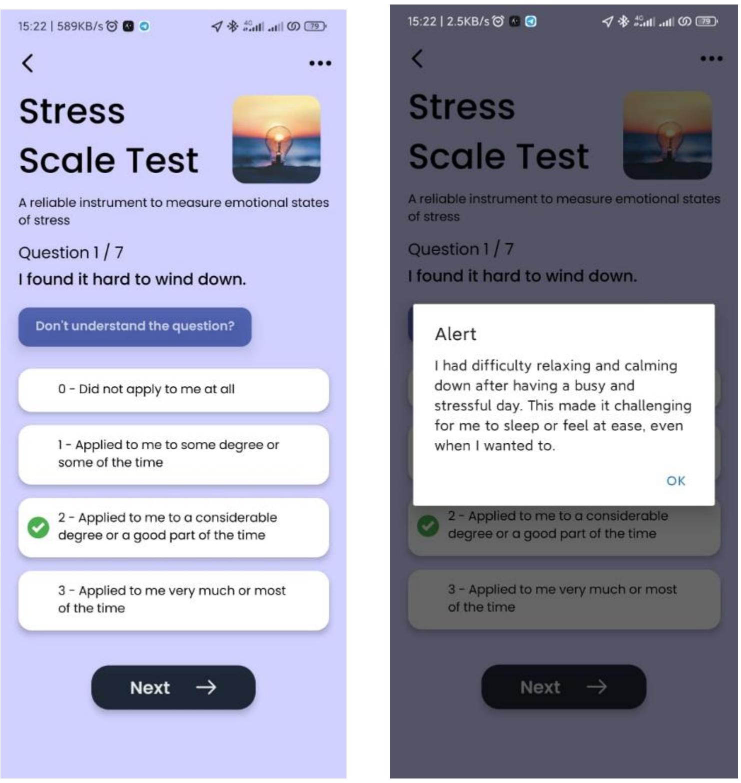
Users are compulsory to answer the DASS-21 questionnaire to understand their condition better so they can choose to carry out the program which suits themselves the best to maintain their mental health. Therefore, the application is programmed to ensure the users need to answer to the stress assessment session at least once to obtain a detailed report. There are total of 21 questions in DASS-21, which specifies the three mental conditions (depression, anxiety, and stress). Since the proposed application only focuses on tackling the stress issues faced by the target users, only seven questions related to stress condition will be provided as shown in Appendix 1. This allows the users to save more time on answering the questionnaire to get the most accurate result.


File Edit Selection View Go Run Terminal Help
Questionnaire.js - LittleChillingHelp - Visual Studio Code
App.js M
Home.js U
MBSRHome.js U
Questionnaire.js U
config.js U
ForgotPassword.js U
Register.js U
Report.js U
QuickDestress.js U
screens > Questionnaire.js > Questionnaire > questionData > options
master*
0 0
React Native Packager
JS Preview Available
Ln 51, Col 65 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

```
const Questionnaire = () => {
  const questionData = [
    {
      id: "1",
      question: "I found it hard to wind down.",
      options: [
        { option: "0 - Did not apply to me at all", points: 0 },
        {
          option: "1 - Applied to me to some degree or some of the time",
          points: 1,
        },
        {
          option:
            "2 - Applied to me to a considerable degree or a good part of the time",
          points: 2,
        },
        {
          option: "3 - Applied to me very much or most of the time",
          points: 3,
        },
      ],
      explanation:
        "I had difficulty relaxing and calming down after having a busy and stressful day. This made it challenging for me to sleep or feel at ease, even when I wanted to
    },
    {
      id: "2",
      question: "I tended to over-react to situation",
      options: [
        { option: "0 - Did not apply to me at all", points: 0 },
        {
          option: "1 - Applied to me to some degree or some of the time",
          points: 1,
        },
        {
          option:
            "2 - Applied to me to a considerable degree or a good part of the time",
          points: 2,
        },
        {
          option: "3 - Applied to me very much or most of the time",
```

Figure 5.6 Code snippet of the seven questions from DASS-21

In React Native, an array list was created to store the data of all questions which includes the questions, options list and points, description of the questions, and the ID of the questions. The question array will be mapped out and listed one by one, using the question ID as the key to navigate through the list. When users answer the questionnaire, they need to select the one answer options where the scores are ranging from 0 to 3. Next question will be displayed until the final question when users click the “Next” button. Besides that, users can click to view the explanation of the question if they do not understand the question as some of the questions consist of wordings which are difficult to understand. A dialog box will prompt to the users and show a simpler meaning of the question. After users answer the last question of the questionnaire, the backend services will calculate the total score of the response and update the score to database with the current timestamp.



Stress Scale Test

A reliable instrument to measure emotional states of stress

Question 1 / 7

I found it hard to wind down.

Don't understand the question?

0 - Did not apply to me at all

1 - Applied to me to some degree or some of the time

2 - Applied to me to a considerable degree or a good part of the time

3 - Applied to me very much or most of the time

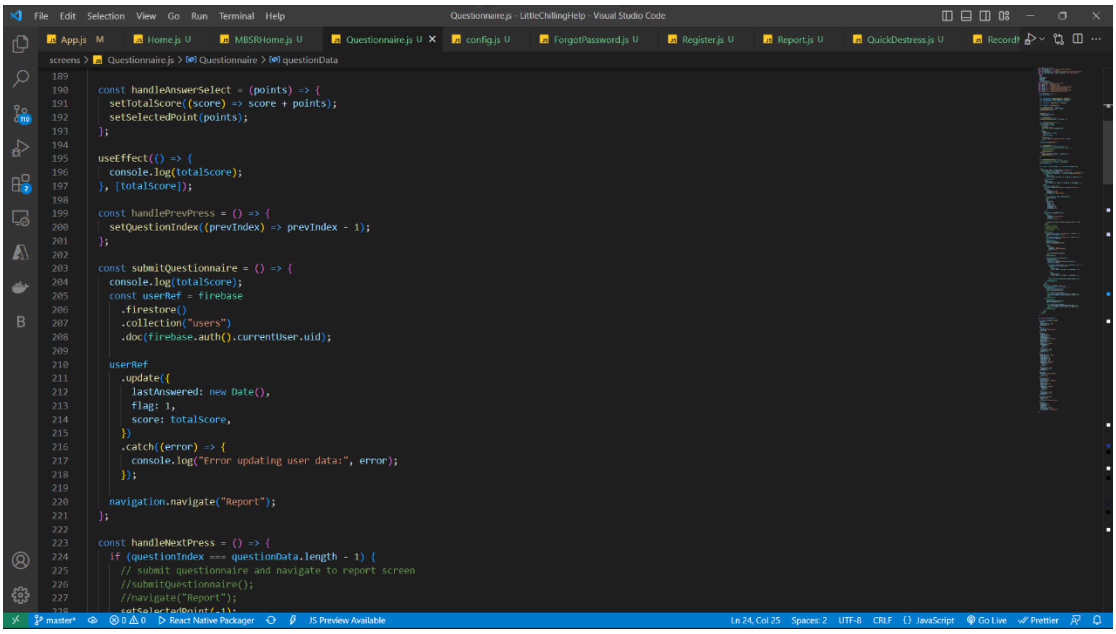
Next

Alert

I had difficulty relaxing and calming down after having a busy and stressful day. This made it challenging for me to sleep or feel at ease, even when I wanted to.

OK

Figure 5.7 DASS-21 questionnaire interface and the explanation of the questions



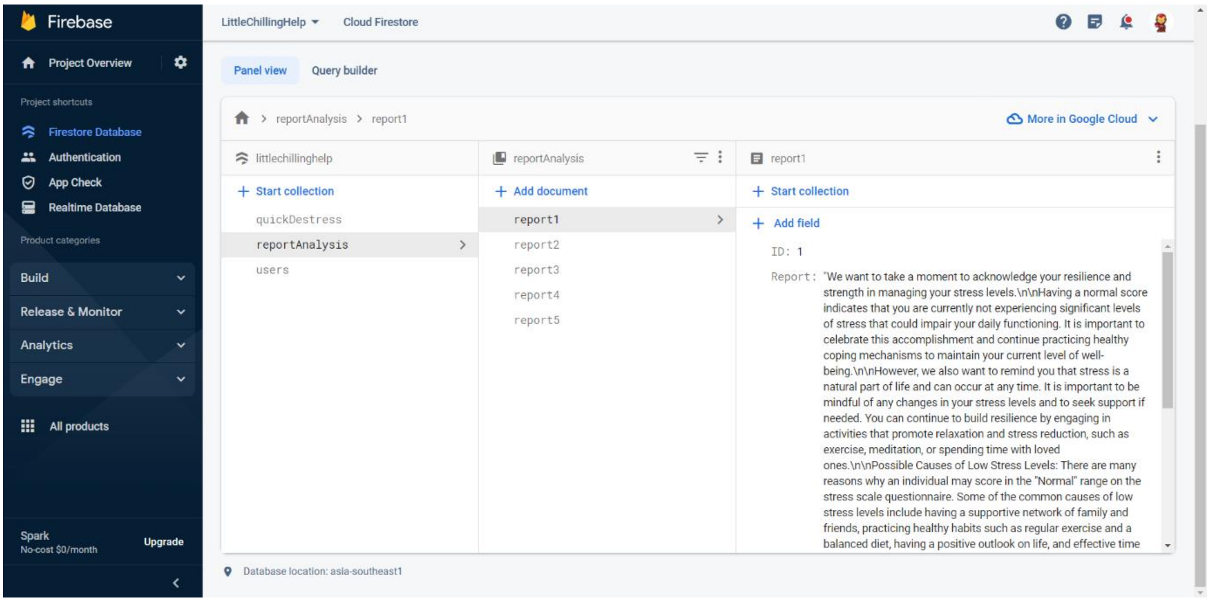
```

const handleAnswerSelect = (points) => {
  setTotalScore((score) => score + points);
  setSelectedPoint(points);
};

useEffect(() => {
  console.log(totalScore);
}, [totalScore]);

const handlePrevPress = () => {
  setQuestionIndex((prevIndex) => prevIndex - 1);
};

const submitQuestionnaire = () => {
  console.log(totalScore);
  const userRef = firebase
    .firestore()
    .collection("users")
    .doc(firebase.auth().currentUser.uid);

  userRef
    .update({
      lastAnswered: new Date(),
      flag: 1,
      score: totalScore,
    })
    .catch((error) => {
      console.log("Error updating user data:", error);
    });

  navigation.navigate("Report");
};

const handleNextPress = () => {
  if (questionIndex === questionData.length - 1) {
    // submit questionnaire and navigate to report screen
    //submitQuestionnaire();
    //navigate("Report");
    setSelectedPoint(-1);
```

Figure 5.8 Code snippet of updating scores to the database

Based on the Figure 5.8, line 203 to 221 indicate the steps to upload total score of the questionnaire to the database. “lastAnswered” will be updated with the current timestamp, the score will be updated too, and most importantly, the “flag” will be changed to 1 from 0 to indicate that the users have once answered to this questionnaire, so users can choose freely whether they still want to answer to the questionnaire or not within a short period of time. They are not compulsory to take part to the assessment again within 5 days after answering the questionnaire.

5.3 Generate report analysis from the database

Once the users completed the DASS-21 questionnaire, then they shall receive a report which provide in depth analysis of their current stress condition. The reports are prepared and stored in the database earlier with unique report ID.

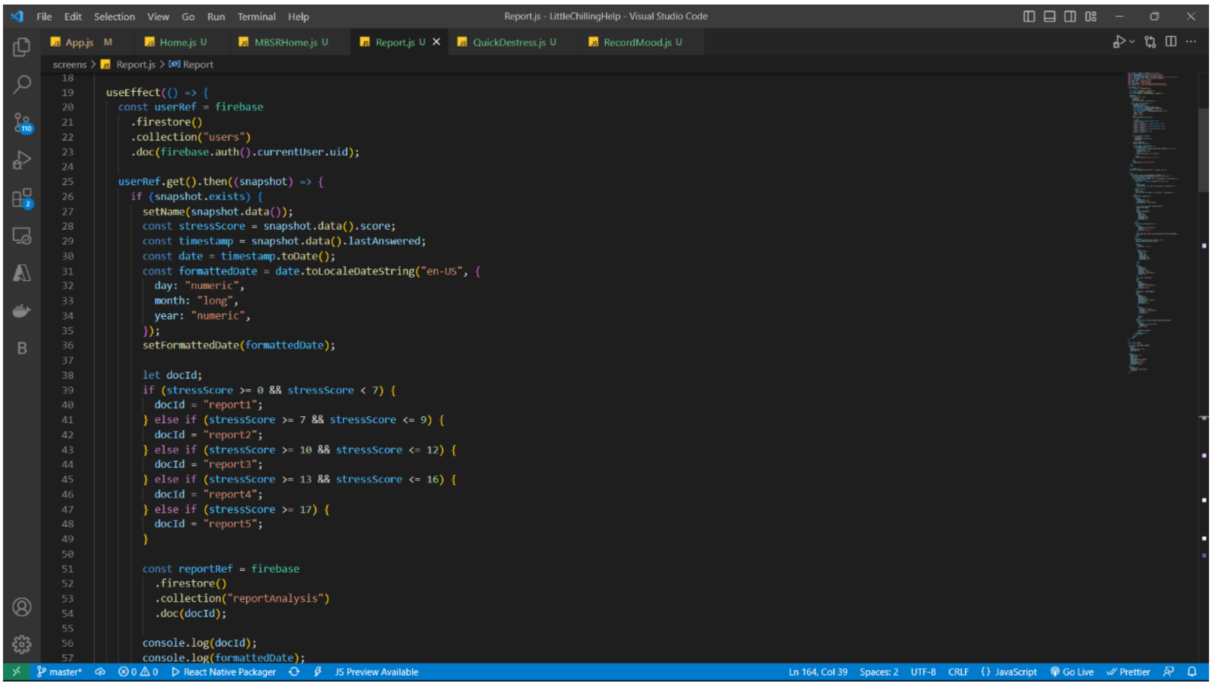


ID: 1

Report: "We want to take a moment to acknowledge your resilience and strength in managing your stress levels.\n\nHaving a normal score indicates that you are currently not experiencing significant levels of stress that could impair your daily functioning. It is important to celebrate this accomplishment and continue practicing healthy coping mechanisms to maintain your current level of well-being.\n\nHowever, we also want to remind you that stress is a natural part of life and can occur at any time. It is important to be mindful of any changes in your stress levels and to seek support if needed. You can continue to build resilience by engaging in activities that promote relaxation and stress reduction, such as exercise, meditation, or spending time with loved ones.\n\nPossible Causes of Low Stress Levels: There are many reasons why an individual may score in the "Normal" range on the stress scale questionnaire. Some of the common causes of low stress levels include having a supportive network of family and friends, practicing healthy habits such as regular exercise and a balanced diet, having a positive outlook on life, and effective time

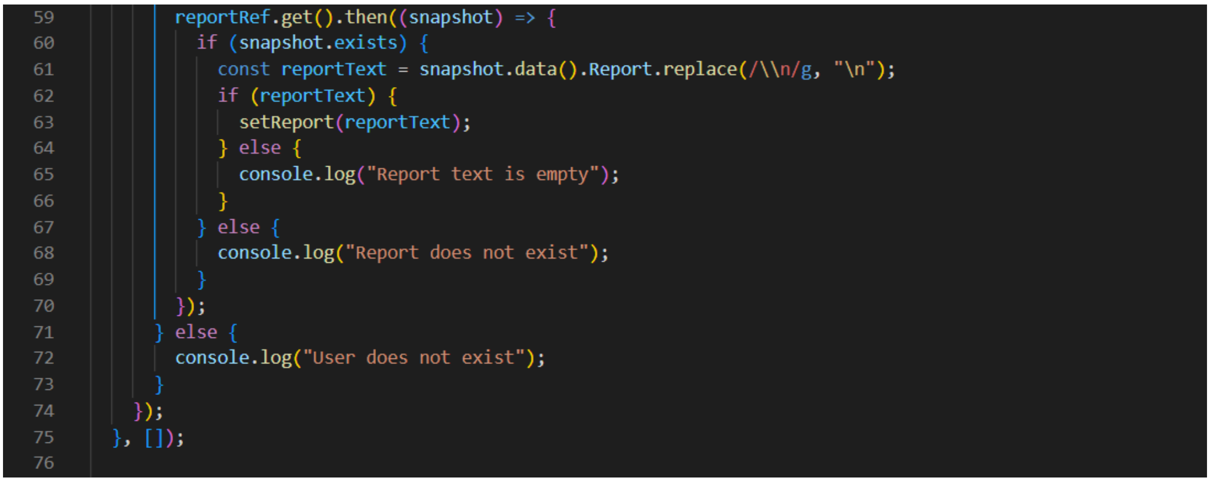
Figure 5.9 List of reports of DASS-21 stress score

As shown in the Figure 5.9, the report list is prepared based on different stress levels. There are total of five stress score levels ranging from normal to extremely severe. Firstly, the application read the user score and the “lastAnswered” from the database to find the recent time of the users answer to the questionnaire. The “lastAnswered” will be convert to readable format in React Native under the “en-gb” format. Based on the Figure 4.10, the “useEffect” library is implemented to carry out the function to retrieve user’s data and report from the database. After the total score of the user is identified, line 38 to 49 used an if-else statement used to filter the score levels to fetch the correct report ID.



```
  useEffect(() => {
    const userRef = firebase
      .firestore()
      .collection("users")
      .doc(firebase.auth().currentUser.uid);

    userRef.get().then((snapshot) => {
      if (snapshot.exists) {
        setName(snapshot.data());
        const stressScore = snapshot.data().score;
        const timestamp = snapshot.data().lastAnswered;
        const date = timestamp.toDate();
        const formattedDate = date.toLocaleDateString("en-US", {
          day: "numeric",
          month: "long",
          year: "numeric",
        });
        setFormattedDate(formattedDate);

        let docId;
        if (stressScore >= 0 && stressScore < 7) {
          docId = "report1";
        } else if (stressScore >= 7 && stressScore <= 9) {
          docId = "report2";
        } else if (stressScore >= 10 && stressScore <= 12) {
          docId = "report3";
        } else if (stressScore >= 13 && stressScore <= 16) {
          docId = "report4";
        } else if (stressScore >= 17) {
          docId = "report5";
        }

        const reportRef = firebase
          .firestore()
          .collection("reportAnalysis")
          .doc(docId);

        console.log(docId);
        console.log(formattedDate);
```

Figure 5.10 Code snippets of the user's data retrieval

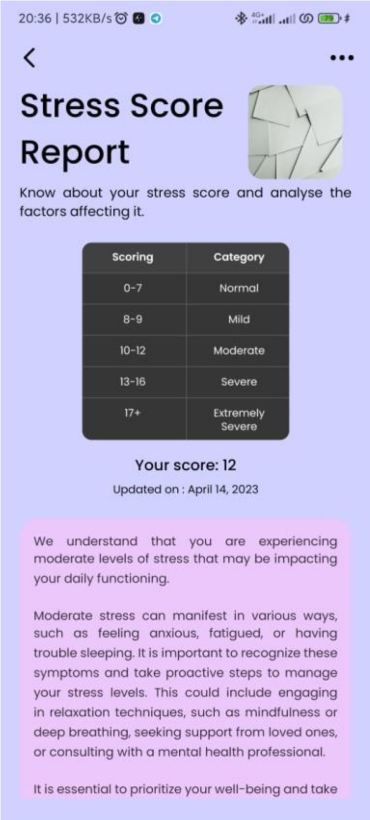


```
59          reportRef.get().then((snapshot) => {
60            if (snapshot.exists) {
61              const reportText = snapshot.data().Report.replace(/\\n/g, "\n");
62              if (reportText) {
63                setReport(reportText);
64              } else {
65                console.log("Report text is empty");
66              }
67            } else {
68              console.log("Report does not exist");
69            }
70          });
71        } else {
72          console.log("User does not exist");
73        }
74      });
75    }, []);
76
```

Figure 5.11 Code snippets of the report retrieval

After the report ID is identified, the "reportRef" will fetch the correct report from the database and show to the users. Line 61 from Figure 5.11 shows that the text in the report is formatted to replace the "\n" word in the report to the actual breaking line so the report is more readable when appeared to the users.

In the application interface, the users will view a table of the stress score categories to understand their categories of the total score. Then, they can read the report which explains the cause of the stress issue and simple recommendation to address the issue in a "<ScrollView>" designed in React Native.



Stress Score Report

Know about your stress score and analyse the factors affecting it.

Scoring	Category
0-7	Normal
8-9	Mild
10-12	Moderate
13-16	Severe
17+	Extremely Severe

Your score: 12

Updated on : April 14, 2023

We understand that you are experiencing moderate levels of stress that may be impacting your daily functioning.

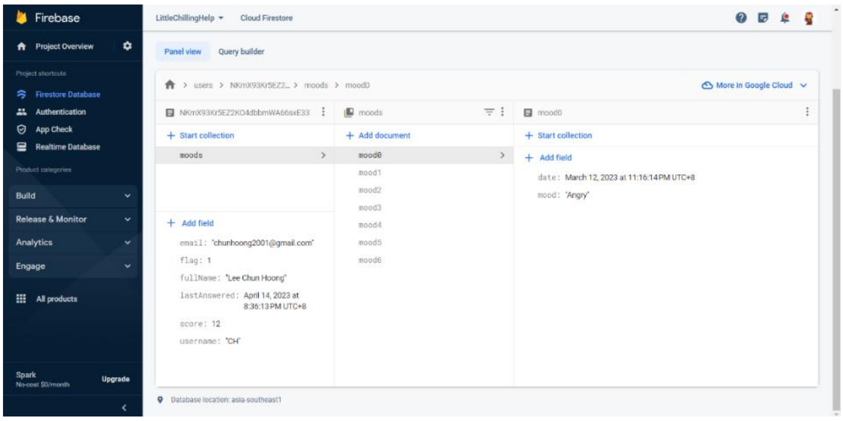
Moderate stress can manifest in various ways, such as feeling anxious, fatigued, or having trouble sleeping. It is important to recognize these symptoms and take proactive steps to manage your stress levels. This could include engaging in relaxation techniques, such as mindfulness or deep breathing, seeking support from loved ones, or consulting with a mental health professional.

It is essential to prioritize your well-being and take

Figure 5.12 User interface of DASS-21 report

5.4 Mood Tracker feature

One of the functions in the proposed application is to allow users to face their true feelings to recognize themselves. Therefore, the mood tracker function is developed to encourage users to record their mood of the day based on different emotions provided as the options. A collection named "moods" is created to store seven moods of a week. This function will display the moods felt for the past seven days, so users will reflect on themselves to learn the why and how do they have such feelings.



moods

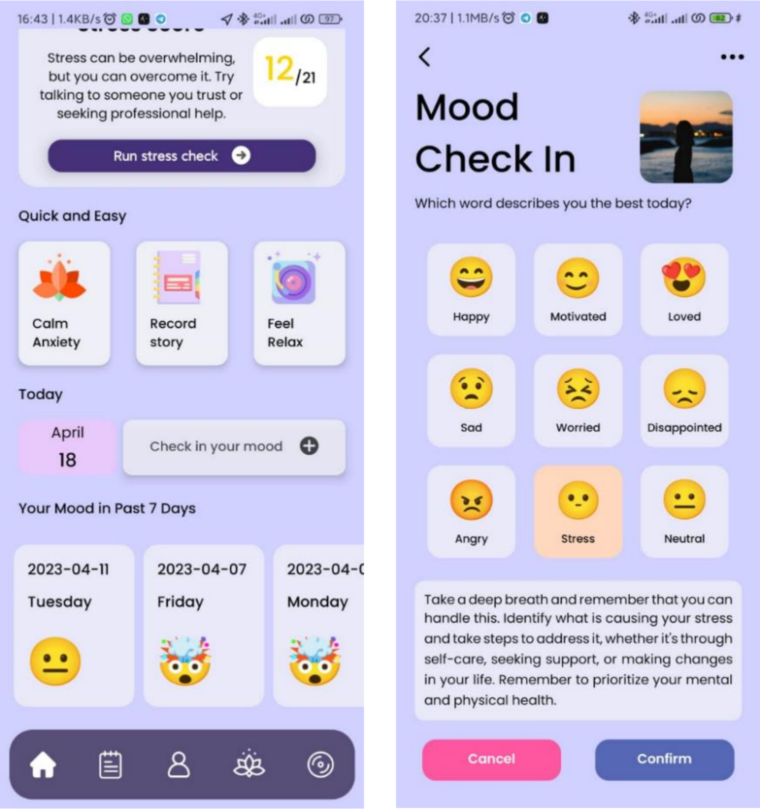
mood0
mood1
mood2
mood3
mood4
mood5
mood6

email: "chunhoong2001@gmail.com"
flag: 1
fullName: "Lee Chun Hoong"
lastAnswered: April 14, 2023 at 8:36:13 PM UTC+8
score: 12
username: "CH"

date: March 12, 2023 at 11:16:14 PM UTC+8
mood: "Angry"

Figure 5.13 Moods collection stored in the Firebase database

Based on the Figure 4.13, each user will record a total of seven moods everyday and the moods can vary from angry, sad, happy, disappointed, stress, neutral and so on. This function is provided at the home screen to allow the users to record their moods without navigating to unnecessary screens.



16:43 | 1.4KB/s

Stress can be overwhelming, but you can overcome it. Try talking to someone you trust or seeking professional help.

12/21

Run stress check

Quick and Easy

Calm Anxiety

Record story

Feel Relax

Today

April 18

Check in your mood

Your Mood in Past 7 Days

2023-04-11 Tuesday

2023-04-07 Friday

2023-04-(Monday

20:37 | 1.1MB/s

Mood Check In

Which word describes you the best today?

Happy

Motivated

Loved

Sad

Worried

Disappointed

Angry

Stress

Neutral

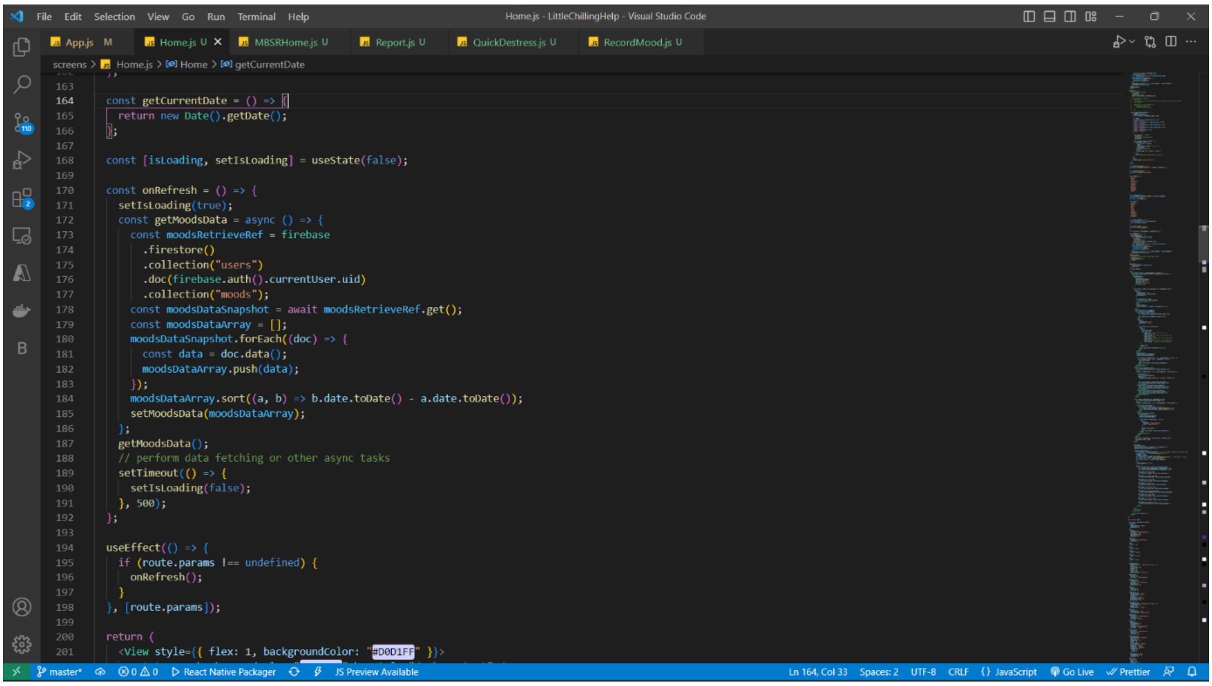
Take a deep breath and remember that you can handle this. Identify what is causing your stress and take steps to address it, whether it's through self-care, seeking support, or making changes in your life. Remember to prioritize your mental and physical health.

Cancel

Confirm

Figure 5.14 Home screen of the application (Left) and Mood Tracker function (Right)

When users click on the check in button, application will navigate to the mood tracker screen to enable them to select mood to be recorded. Users can preview each mood and read the description to understand the feelings better. The descriptions of the moods are stored in an array list to be mapped out through React Native to display all moods and the description when the mood is clicked by the user. Then, user can click the confirm button to save the mood recorded into the database and navigate back to the home screen. The screen will auto refresh to display the updated list of the moods which also specify the day of the week and the date recorded. Here, “useEffect” is used to ensure that the application can be auto refresh every time when the user navigate back to home screen. In the case, that the application is not refreshed automatically, the users can still manually update the home screen by pulling the screen downwards to refresh.



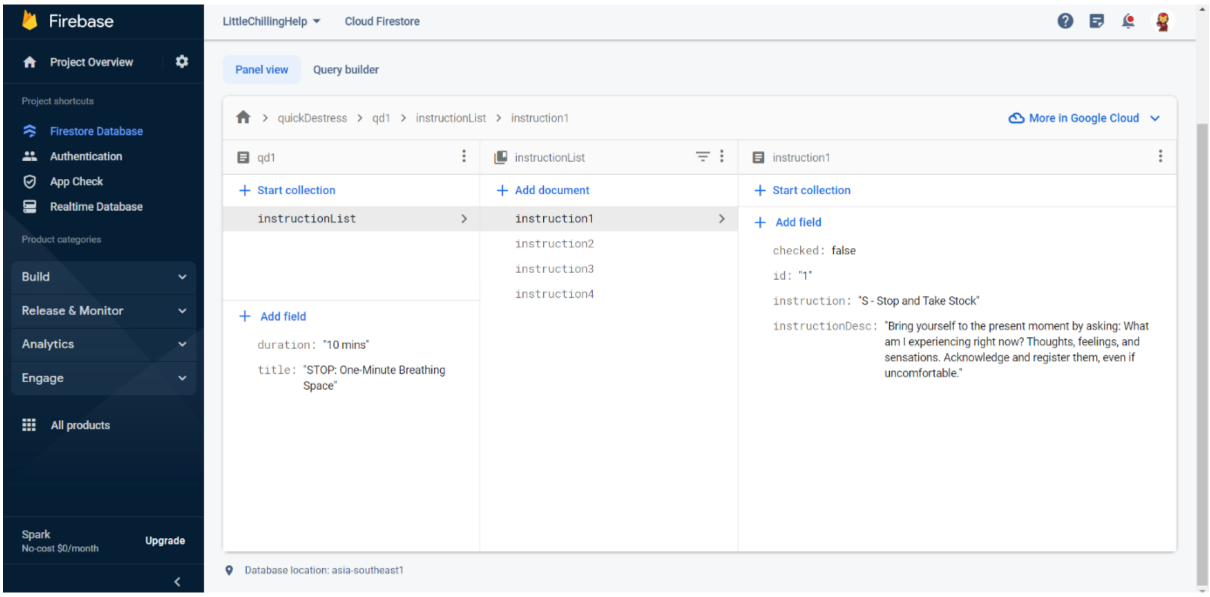
```
163
164  const getCurrentDate = () => {
165    return new Date().getDate();
166  };
167
168  const [isLoading, setIsLoading] = useState(false);
169
170  const onRefresh = () => {
171    setIsLoading(true);
172    const getMoodsData = async () => {
173      const moodsRetrieveRef = firebase
174        .firestore()
175        .collection("users")
176        .doc(firebase.auth().currentUser.uid)
177        .collection("moods");
178      const moodsDataSnapshot = await moodsRetrieveRef.get();
179      const moodsDataArray = [];
180      moodsDataSnapshot.forEach((doc) => {
181        const data = doc.data();
182        moodsDataArray.push(data);
183      });
184      moodsDataArray.sort((a, b) => b.date.toDate() - a.date.toDate());
185      setMoodsData(moodsDataArray);
186    };
187    getMoodsData();
188    // perform data fetching or other async tasks
189    setTimeout(() => {
190      setIsLoading(false);
191    }, 500);
192  };
193
194  useEffect(() => {
195    if (route.params !== undefined) {
196      onRefresh();
197    }
198  }, [route.params]);
199
200  return (
201    <View style={{ flex: 1, backgroundColor: "#D0D1FF" }}>
```

Figure 5.15 Code snippets of the auto refreshing of home screen

The code will identify whether the home screen needs to be auto refreshed or otherwise as shown in the Figure 5.15 above. Line 170 to 192 will call the "onRefresh" function to set the loading variable to true and indicate that the page need auto refresh. Then the application will fetch data from moods collection to display the latest list. The list will be arranged to descending order where the latest date and mood will be displayed at first. All the elements inside the collection will be pushed into an array again in React Native. This is because React Native cannot display a collection of attributes from another site without declaring them into an array in React. When the page is loading to refresh, the time set is to 500 milliseconds to complete the refreshing. At the same time, the frequency of Firebase Firestore database documents read will increase but thanks to the "useEffect" library, the number of documents read and write will not exceed 50K per day due to the 500 milliseconds loading time. Otherwise, the home screen could have auto refresh infinitely and exceed the limit of the Firebase.

5.5 Development of MBSR Quick Destress feature

MBSR program is applied to integrate in the proposed application to help users to reduce their stress, effectively. In FYP1, the MBSR Quick Destress function is developed to provide users a quick solution to reduce stress in a short time. Three Quick Destress activities are provided in the MBSR screen with the activities title and the duration to complete the activity. The program details including the instruction are listed inside the database and will be fetched to the Visual Studio Code by placing all three activities into one array list.



quickDestress > qd1 > instructionList > instruction1

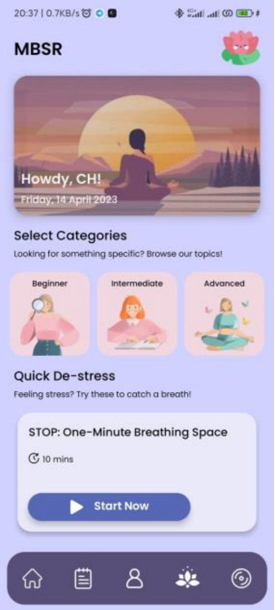
qd1
instructionList
duration: "10 mins"
title: "STOP: One-Minute Breathing Space"

instructionList
instruction1
instruction2
instruction3
instruction4

instruction1
checked: false
id: "1"
instruction: "S - Stop and Take Stock"
instructionDesc: "Bring yourself to the present moment by asking: What am I experiencing right now? Thoughts, feelings, and sensations. Acknowledge and register them, even if uncomfortable."

Database location: asia-southeast1

Figure 5.16 MBSR program instructions list stored in the database


MBSR

Howdy, CH!
Friday, 14 April 2023

Select Categories
Looking for something specific? Browse our topics!

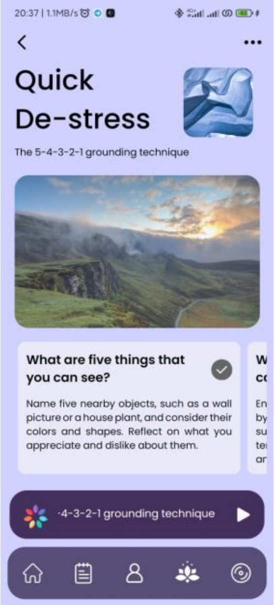
Beginner
Intermediate
Advanced

Quick De-stress
Feeling stress? Try these to catch a breath!

STOP: One-Minute Breathing Space
10 mins
Start Now

Figure 5.17 MBSR Home screen

MBSR Quick Destress program details are stored as one collection and consists of three activities which identified by unique activity ID as shown in the Figure 5.16. Instructions are saved as different documents in one activity collection. Based on the Figure 5.17, the Quick Destress activities are displayed in a vertical “<ScrollView>” so users can scroll through the list and select the activity they want. Besides that, the main activities of MBSR program especially meditation are categorized into three groups, beginner, intermediate, and advanced. This section is prepared for the users who want to perform a mindfulness meditation to reduce their stress level based on the skill levels. This function will be continued for development in FYP2.


20:37 | 1.1MB/s
Quick De-stress
The 5-4-3-2-1 grounding technique

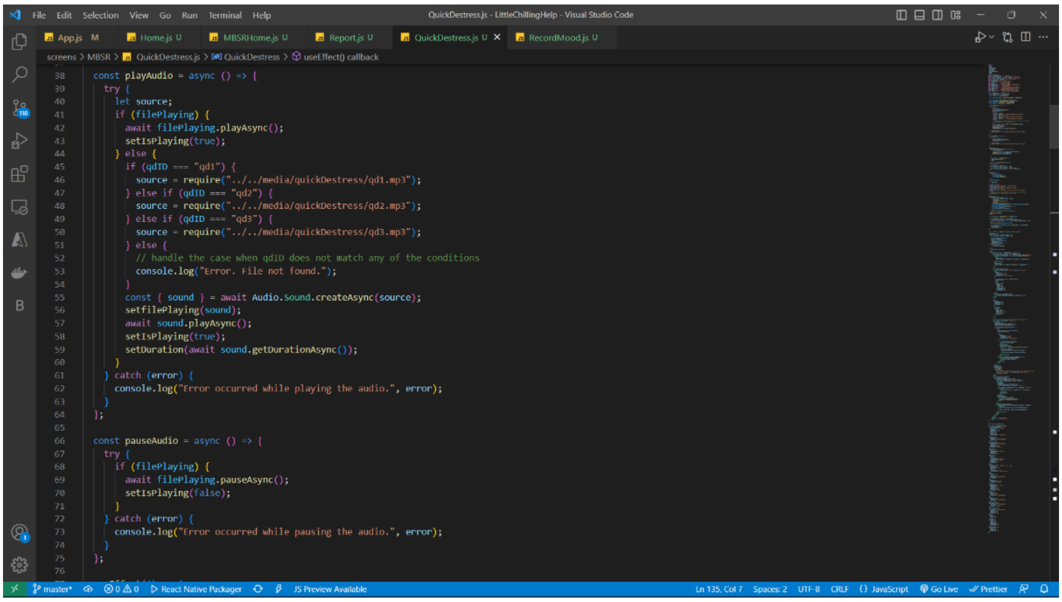
What are five things that you can see?

Name five nearby objects, such as a wall picture or a house plant, and consider their colors and shapes. Reflect on what you appreciate and dislike about them.

-4-3-2-1 grounding technique

Figure 5.18 Quick Destress activity interface – The 5-4-3-2-1 grounding technique

When the users click the “Start Now” button on the activity, they will be navigated to the activity screen to perform the technique. The picture in the middle of the screen is displayed randomly to give the users a feeling of serenity. Users can scroll through the instruction to read them one by one by following the audio instruction provided during the activity as shown in the Figure 5.18 above.



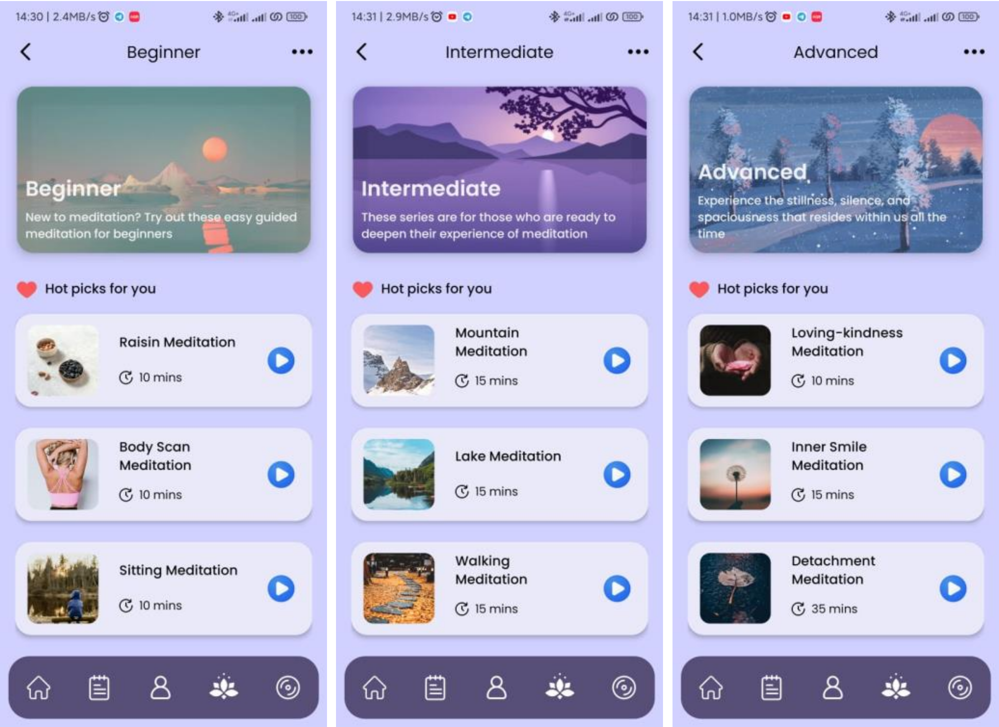
```
38  const playAudio = async () => {
39    try {
40      let source;
41      if (filePlaying) {
42        await filePlaying.playAsync();
43        setIsPlaying(true);
44      } else {
45        if (qdID === "qd1") {
46          source = require("../../media/quickDestress/qd1.mp3");
47        } else if (qdID === "qd2") {
48          source = require("../../media/quickDestress/qd2.mp3");
49        } else if (qdID === "qd3") {
50          source = require("../../media/quickDestress/qd3.mp3");
51        } else {
52          // handle the case when qdID does not match any of the conditions
53          console.log("Error. File not found.");
54        }
55        const { sound } = await Audio.Sound.createAsync(source);
56        setfilePlaying(sound);
57        await sound.playAsync();
58        setIsPlaying(true);
59        setDuration(await sound.getDurationAsync());
60      }
61    } catch (error) {
62      console.log("Error occurred while playing the audio.", error);
63    }
64  };
65
66  const pauseAudio = async () => {
67    try {
68      if (filePlaying) {
69        await filePlaying.pauseAsync();
70        setIsPlaying(false);
71      }
72    } catch (error) {
73      console.log("Error occurred while pausing the audio.", error);
74    }
75  };
76
```

Figure 5.19 Code snippets of the audio instruction being played and paused

The audio instruction can be played and paused at any time by the users. To achieve that, the “Audio” library is imported from “expo-av” to load the audio file when the audio has not been playing yet. Based on the Figure 5.19 line 38 to 64, the code snippets explains that if there is an audio file playing, then the “playAsync” function will run and resume the playing of audio file. If there is no file playing at the current time, then the “source” variable will match with the “qdID” declared earlier and identify the audio file to be played. Next, a new “sound” object will be created by the API using the “createAsync” method and passes the “sound” object as the parameter. Then, the audio file is ready to play now when “isPlaying” is set to true. While to pause the audio from playing, “pauseAudio” function from line 66 to 75 will take place. If there is a file playing, “pauseAsync” function will run and set the “isPlaying” to false.

5.6 Developing of MBSR Meditation Feature

The proposed application is provided with the main functionality of the MBSR Meditation exercise to the user in order to relieve stress by different exercises. The features will refer to the complete MBSR 8-week course and extract most of the meditation exercise ranging from beginner, intermediate, and advanced, which is according to the mastery of mindfulness [30]. Each meditation exercise details are stored in the firebase database in different collection so each time the user wants to select one of the categories, the information is fetched from the database according to the category ID first. Therefore, each category will present 3 meditation exercises with different duration as shown as the Figure 5.20 below. Each meditation exercise will present its title and duration needed to complete the exercise.



14:30 | 2.4MB/s

Beginner

Beginner

New to meditation? Try out these easy guided meditation for beginners

Hot picks for you

Raisin Meditation 10 mins

Body Scan Meditation 10 mins

Sitting Meditation 10 mins

14:31 | 2.9MB/s

Intermediate

Intermediate

These series are for those who are ready to deepen their experience of meditation

Hot picks for you

Mountain Meditation 15 mins

Lake Meditation 15 mins

Walking Meditation 15 mins

14:31 | 1.0MB/s

Advanced

Advanced

Experience the stillness, silence, and spaciousness that resides within us all the time

Hot picks for you

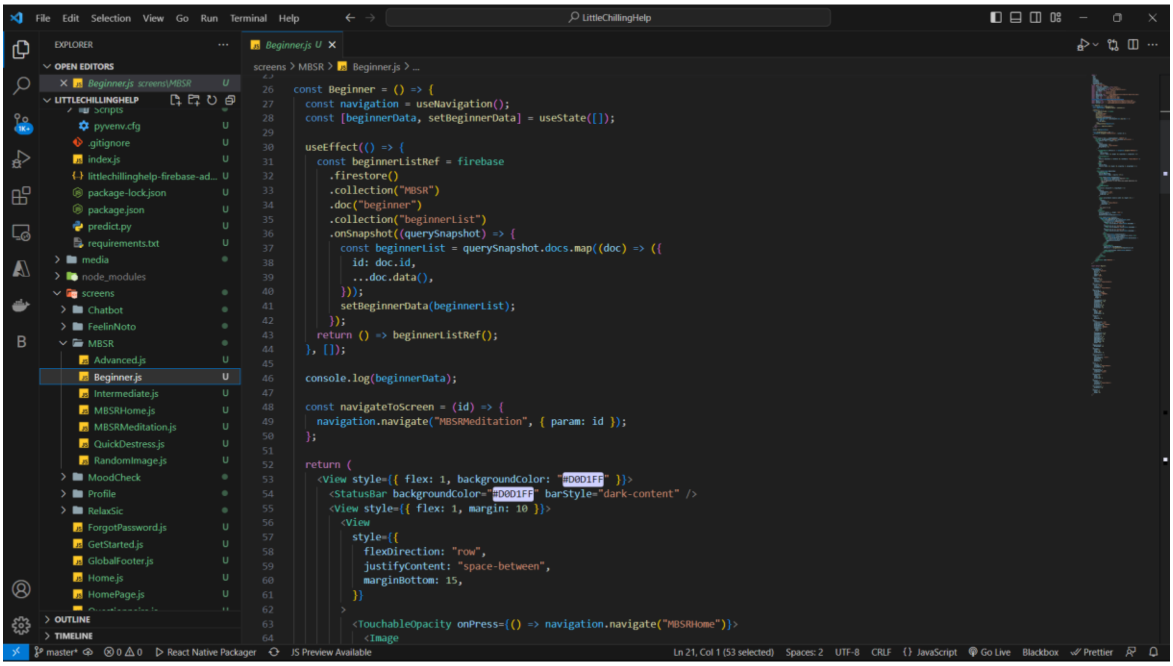
Loving-kindness Meditation 10 mins

Inner Smile Meditation 15 mins

Detachment Meditation 35 mins

Figure 5.20 List of meditation exercises of all three categories

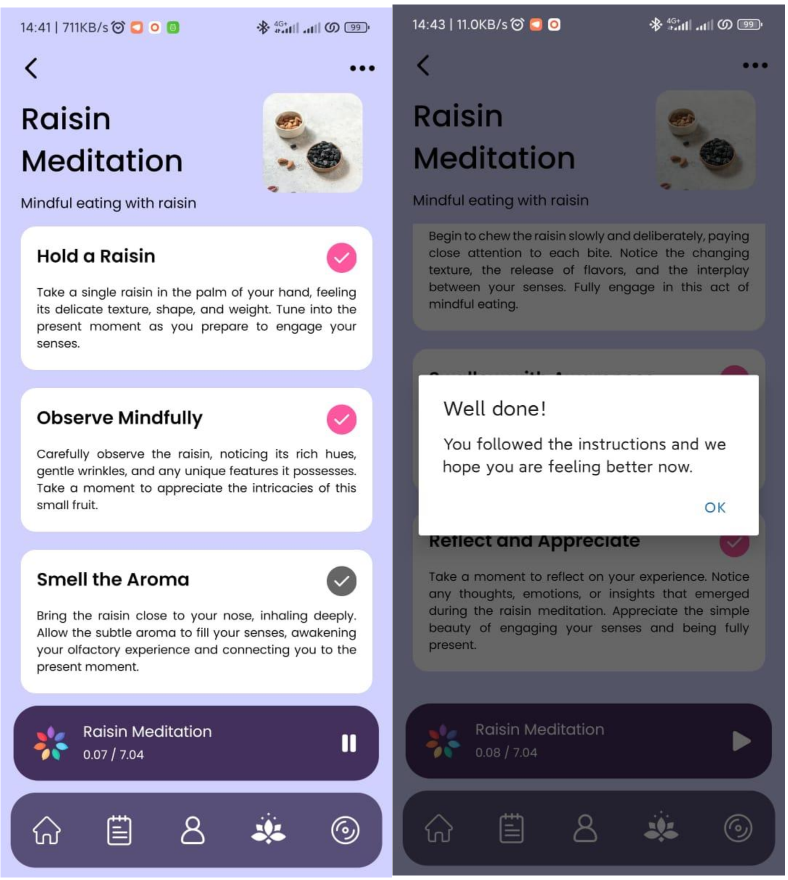
All these information is fetched from the database and display according to the user response. When the users finally select one meditation exercise to carry out, the system will collect the meditation ID selected and match it with the database, so the system will direct the users to another page to start the mediation exercise. As shown in Figure 5.21 below, if the users select the “Beginner” category for meditation exercises, the system will recognize it as a beginner-type meditation. It will then link to the Firebase database and display all available meditation exercises under the “BeginnerList” collection, organized by their meditation ID. Thus, when the users select a meditation exercise from the “Beginner” category, the system will redirect them to the “MBSRMeditation” page with the corresponding ID reference.



```
const Beginner = () => {
  const navigation = useNavigation();
  const [beginnerData, setBeginnerData] = useState([]);

  useEffect(() => {
    const beginnerListRef = firebase
      .firestore()
      .collection("MBSR")
      .doc("beginner")
      .collection("beginnerList")
      .onSnapshot((querySnapshot) => {
        const beginnerList = querySnapshot.docs.map((doc) => ({
          id: doc.id,
          ...doc.data(),
        }));
        setBeginnerData(beginnerList);
      });
    return () => beginnerListRef();
  }, []);

  console.log(beginnerData);

  const navigateToScreen = (id) => {
    navigation.navigate("MBSRMeditation", { param: id });
  };

  return (
    <View style={{ flex: 1, backgroundColor: "#D0D1FF" }}>
      <StatusBar backgroundColor="#D0D1FF" barStyle="dark-content" />
      <View style={{ flex: 1, margin: 10 }}>
        <View
          style={{
            flexDirection: "row",
            justifyContent: "space-between",
            marginBottom: 15,
          }}
        >
          <TouchableOpacity onPress={() => navigation.navigate("MBSRHome")}>
            <Image
```

Figure 5.21 Code snippet for retrieving information about meditation exercises

After successfully identify the meditation exercise in the database, the system will display all the instructions in a new page. Expo Audio library is used to implement the audio task created in the application. It is because the meditation exercise will provide both the text and audio instruction, Every time the users select one mediation exercise, the audio file will be fetched from the directory by the Expo Audio library This is because the library can offer different function to control the audio task. First, the audio file is loaded into the application function. Figure 5.22 below shows the complete page when the users are carrying out an MBSR meditation session. When users click the play/pause button on the audio interface, the play/pause function will be called, and the audio file will be played based on the playback status. Expo Audio library can also calculate the duration of the audio and the current time track of the audio to display the audio duration, It is calculated in milliseconds first and then converted to minutes and seconds. While users are conducting the meditation exercise, user can read the text instruction. When they complete one instruction, they can mark or label the instruction as checked, and until the instructions are all checked. This indicates that the users have completed the meditation as they reach the end of the instruction. Moreover by the time audio file has stop playing, it means the meditation exercise has completed. Therefore, when the conditions in the program are met, where the music stop playing and user completes all the instruction, the system will pop up a dialog box saying that the users have completed the meditation exercise.



Raisin Meditation

Mindful eating with raisin

Hold a Raisin

Take a single raisin in the palm of your hand, feeling its delicate texture, shape, and weight. Tune into the present moment as you prepare to engage your senses.

Observe Mindfully

Carefully observe the raisin, noticing its rich hues, gentle wrinkles, and any unique features it possesses. Take a moment to appreciate the intricacies of this small fruit.

Smell the Aroma

Bring the raisin close to your nose, inhaling deeply. Allow the subtle aroma to fill your senses, awakening your olfactory experience and connecting you to the present moment.

Raisin Meditation
0.07 / 7.04

Raisin Meditation

Mindful eating with raisin

Begin to chew the raisin slowly and deliberately, paying close attention to each bite. Notice the changing texture, the release of flavors, and the interplay between your senses. Fully engage in this act of mindful eating.

Well done!

You followed the instructions and we hope you are feeling better now.

Reflect and Appreciate

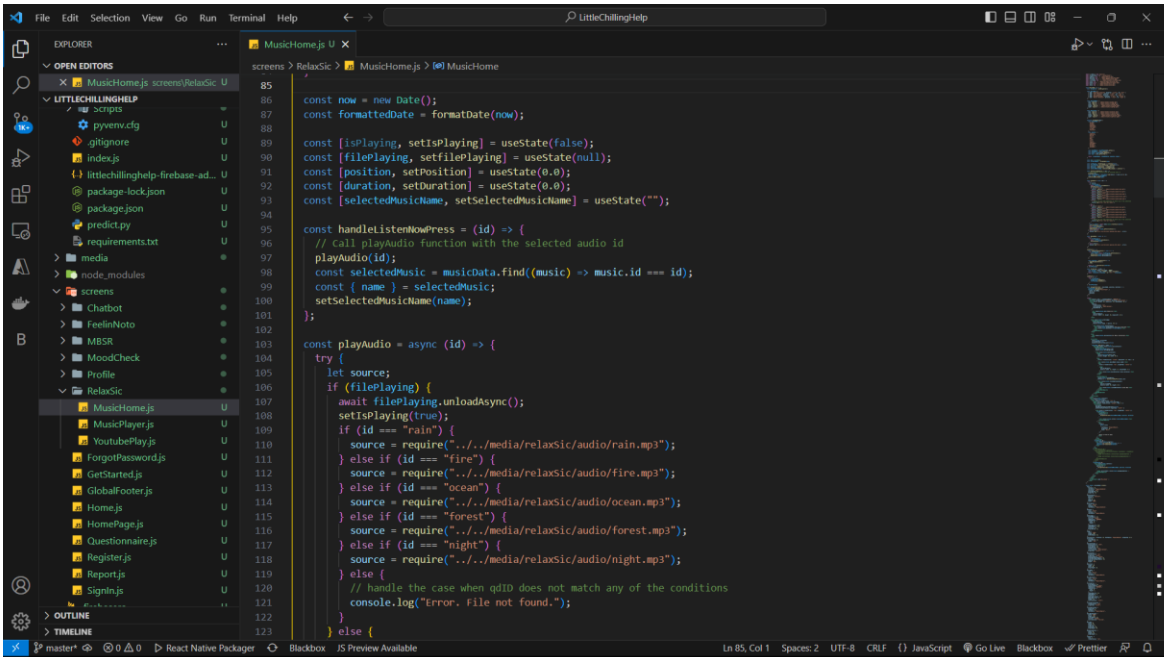
Take a moment to reflect on your experience. Notice any thoughts, emotions, or insights that emerged during the raisin meditation. Appreciate the simple beauty of engaging your senses and being fully present.

Raisin Meditation
0.08 / 7.04

Figure 5.22 Complete page displayed during the meditation exercise

5.7 Designing a Music Player for MBSR Relaxing Music

A music player with controllable functions is created in the proposed application to display to play MBSR relaxing calming music to the user. The music will be displayed to the user as a playlist for user to choose to listen. The MBSR music playlist consists of different calming music from places and weather such as ocean music, fire music, forest music, rain music, and many more. Each music is labelled with a unique music ID or a key to differentiate them in the collection list. Hence the whole music playlist will display all music available according to the music ID/key. Besides that, the ID of the music will link to the related information and retrieve the related information about the individual music. Based on Figure 5.23, the process of loading the audio file onto the task is demonstrated, preparing it for playback.

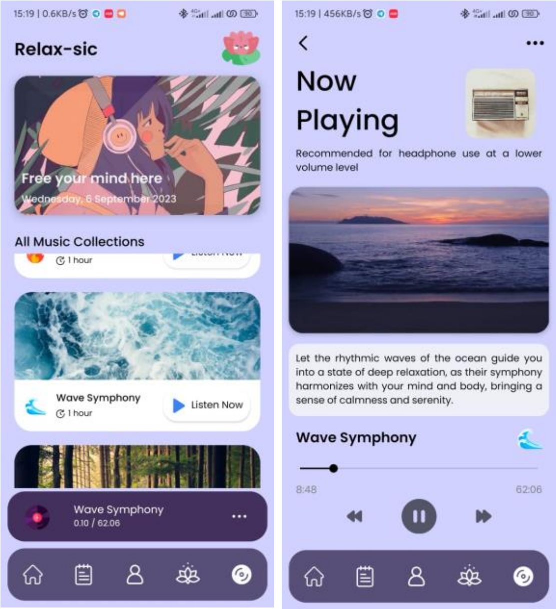


```
const now = new Date();
const formattedDate = formatDate(now);

const [isPlaying, setIsPlaying] = useState(false);
const [filePlaying, setfilePlaying] = useState(null);
const [position, setPosition] = useState(0.0);
const [duration, setDuration] = useState(0.0);
const [selectedMusicName, setSelectedMusicName] = useState("");

const handleListenNowPress = (id) => {
  // Call playAudio function with the selected audio id
  playAudio(id);
  const selectedMusic = musicData.find((music) => music.id === id);
  const { name } = selectedMusic;
  setSelectedMusicName(name);
};

const playAudio = async (id) => {
  try {
    let source;
    if (filePlaying) {
      await filePlaying.unloadAsync();
      setIsPlaying(true);
      if (id === "rain") {
        source = require("../../media/relaxSic/audio/rain.mp3");
      } else if (id === "fire") {
        source = require("../../media/relaxSic/audio/fire.mp3");
      } else if (id === "ocean") {
        source = require("../../media/relaxSic/audio/ocean.mp3");
      } else if (id === "forest") {
        source = require("../../media/relaxSic/audio/forest.mp3");
      } else if (id === "night") {
        source = require("../../media/relaxSic/audio/night.mp3");
      } else {
        // handle the case when qdID does not match any of the conditions
        console.log("Error. File not found.");
      }
    } else {
```

Figure 5.23 Code snippet of loading the audio file to prepare the music playing

The music playlist includes the music name, cover image of the music, music duration, and music type. When user clicks the “Play Now” button, the music will be played according to the music selected. The music player is mimicking the Spotify Music Player interface. When the certain music is selected by the user, there will be a mini music player bar to show the current music playing with play/pause button. In order to view the full complete music player in one page, user can click on the mini bar music player to be directed to the new page as shown as the Figure 5.24 below.

In the new page, Expo Audio Library is implemented once again to load the audio file. Play/Pause button is designed with the related function based on the music playback current status. However, to design the page inspired by Spotify Music Player, some additional functions are added. These include a new music slider for track progressing, and fast forward and fast backward the music for 30 seconds. Besides that, music video in mp4 format is also attached in the application because it can bring more calm feeling by watching the video. In order to integrate related music video, Expo Video Library is implemented to attach the video to the music player interface. Therefore, at the same time, user can listen to the music to calm their inner self, reduce stress, while also watching the music video to get the same effect of calming self-down.



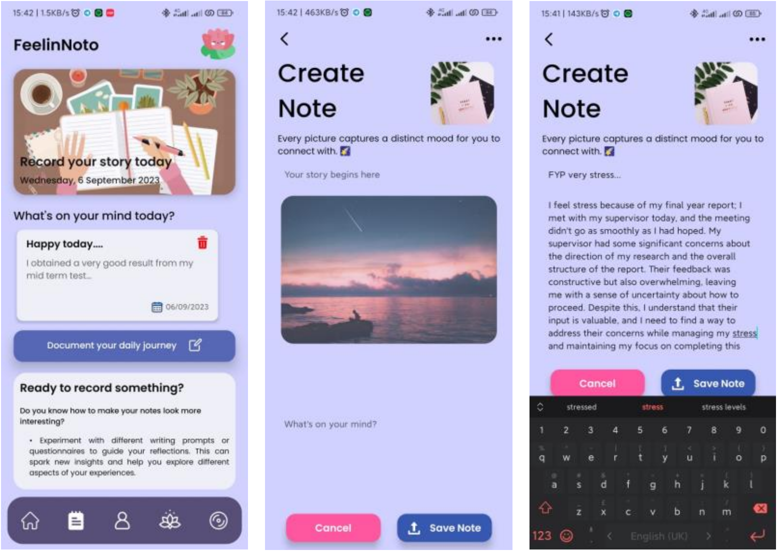
Recommended for headphone use at a lower volume level

Let the rhythmic waves of the ocean guide you into a state of deep relaxation, as their symphony harmonizes with your mind and body, bringing a sense of calmness and serenity.

Figure 5.24 MBSR relaxing music player interface

5.8 Note Taking feature as Journaling Tool

The note taking feature is designed to allow the user to document their story of the day, what mood they experience, and the events happened throughout the day. It is similar like a self-diary, but it is more focused on the thoughts and feeling experienced by each user. Figure 5.25 below represents the journaling tool application interface from creating a note to saving a note.



I obtained a very good result from my mid term test...

Ready to record something?

Do you know how to make your notes look more interesting?

- Experiment with different writing prompts or questionnaires to guide your reflections. This can spark new insights and help you explore different aspects of your experiences.

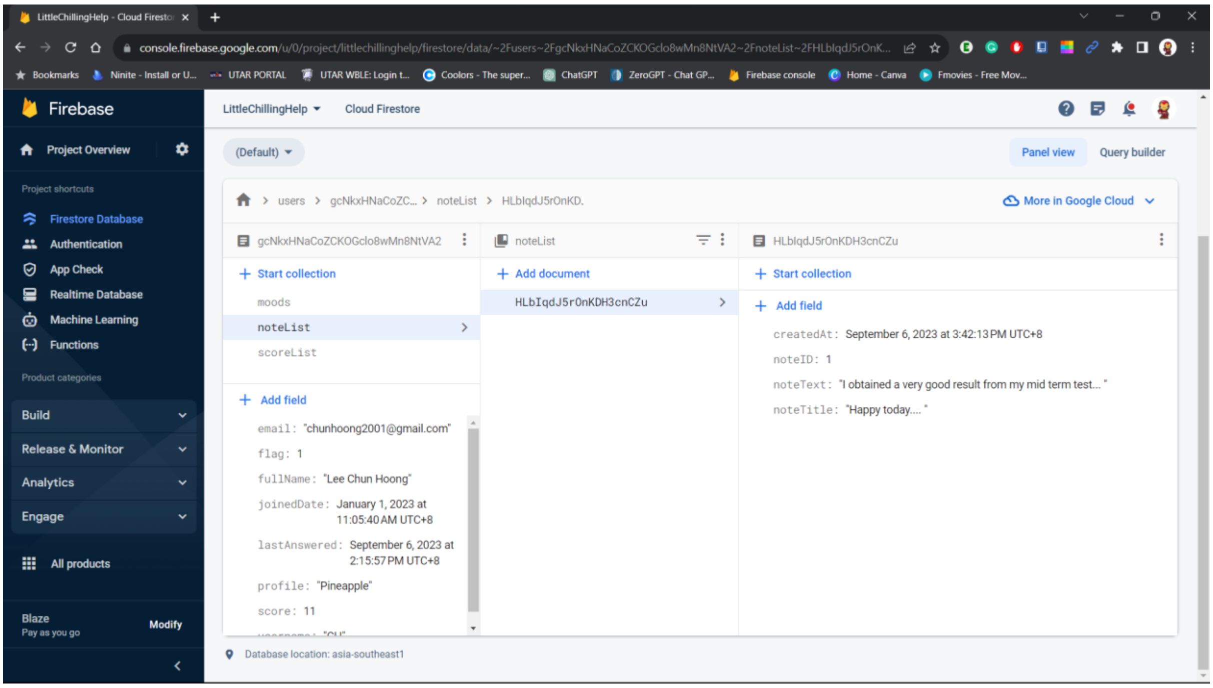
Every picture captures a distinct mood for you to connect with. 🌠

FYP very stress...

I feel stress because of my final year report; I met with my supervisor today, and the meeting didn't go as smoothly as I had hoped. My supervisor had some significant concerns about the direction of my research and the overall structure of the report. Their feedback was constructive but also overwhelming, leaving me with a sense of uncertainty about how to proceed. Despite this, I understand that their input is valuable, and I need to find a way to address their concerns while managing my stress and maintaining my focus on completing this

Figure 5.25 Journaling tool application interface

The feature is developed by storing every created note in the user directory in firebase database collection. This indicates that all notes are linked under the user account, so that it will not share the same repository with same active users, Otherwise, the database consistency and integrity will be affected. Moreover, before the users start to create the note, they can view the existing notes created earlier and saved in the database. According to the Figure 5.26, the notes are stored in the collection named "noteList" and all the notes are arranged in descending order by displaying the latest notes first. The note creation function is developed if users want to create a new note to document their mood and thoughts of the day. When creating each note, note title will be set as a headline of the story or title of the story, then users will proceed to write the note description to further describe their story of the day. The text will then be saved inside the database, with attribute note title, note description, and created timestamp.



gcNkxHNaCoZCKOGclo8wMn8NtVA2

moods
noteList
scoreList

email: "chunhoong2001@gmail.com"
flag: 1
fullName: "Lee Chun Hoong"
joinedDate: January 1, 2023 at 11:05:40 AM UTC+8
lastAnswered: September 6, 2023 at 2:15:57 PM UTC+8
profile: "Pineapple"
score: 11

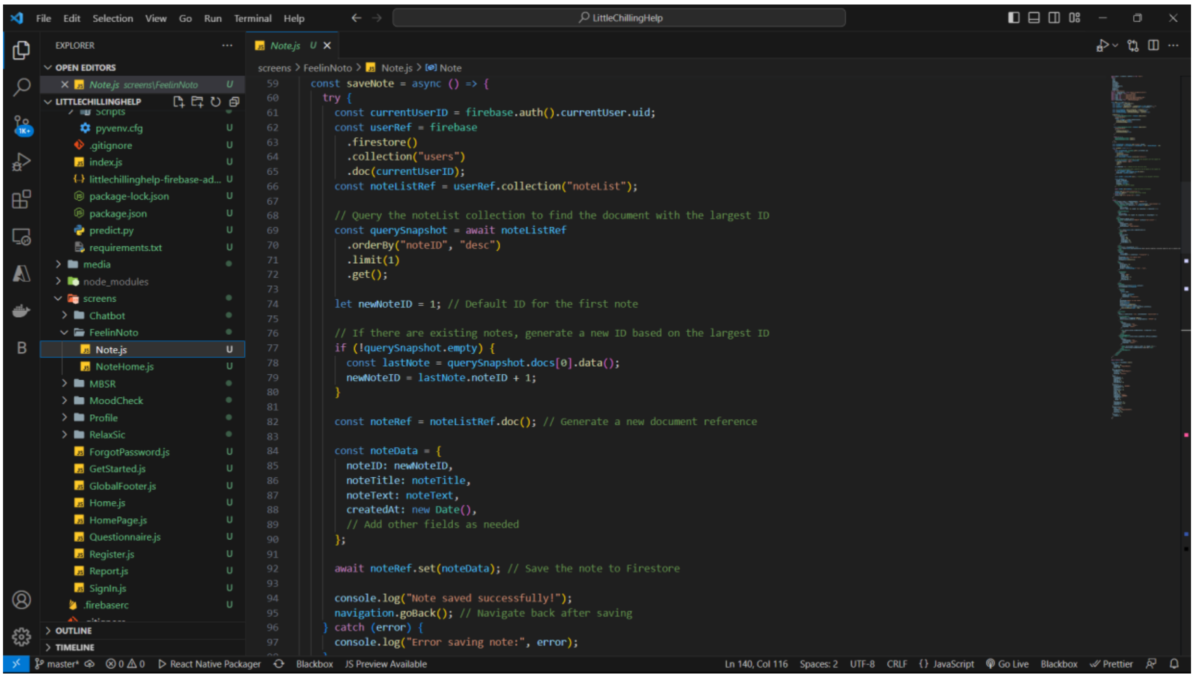
noteList

HLbIqdJ5rOnKDH3cnCZu

createdAt: September 6, 2023 at 3:42:13 PM UTC+8
noteID: 1
noteText: "I obtained a very good result from my mid term test... "
noteTitle: "Happy today.... "

Figure 5.26 "noteList" collection in the Firebase database to store the user's notes

Figure 5.27 below demonstrates the process of creating a new note within the application and subsequently saving it to the Firebase database. When the note is successfully created in the database, user will be redirected to the homepage and the displayed note will be auto refreshed to display the latest created notes. Besides that, the edit of the note function also available to allow users to edit the existing notes. Users can click on the selected notes and enter the note typing editing field page. The note list in the database is then refreshed with the latest updates to allow user to view the recent changes. Not only that, in every note displayed on the page, user can click on the red rubbish bin icon to delete a note from the database in order to remove the notes. A note can be successfully removed after the confirmation from the user. Then the note list will be refreshed again and display the newly changed list.



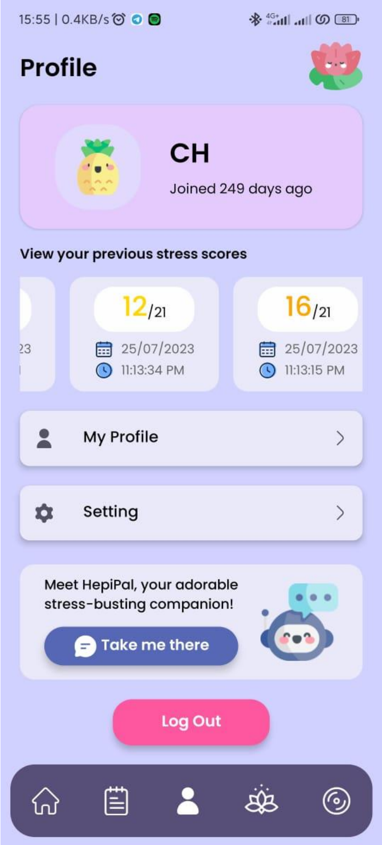
```
const saveNote = async () => {
  try {
    const currentUserID = firebase.auth().currentUser.uid;
    const userRef = firebase
      .firestore()
      .collection("users")
      .doc(currentUserID);
    const noteListRef = userRef.collection("noteList");

    // Query the noteList collection to find the document with the largest ID
    const querySnapshot = await noteListRef
      .orderBy("noteID", "desc")
      .limit(1)
      .get();

    let newNoteID = 1; // Default ID for the first note

    // If there are existing notes, generate a new ID based on the largest ID
    if (!querySnapshot.empty) {
      const lastNote = querySnapshot.docs[0].data();
      newNoteID = lastNote.noteID + 1;
    }

    const noteRef = noteListRef.doc(); // Generate a new document reference

    const noteData = {
      noteID: newNoteID,
      noteTitle: noteTitle,
      noteText: noteText,
      createdAt: new Date(),
      // Add other fields as needed
    };

    await noteRef.set(noteData); // Save the note to Firestore

    console.log("Note saved successfully!");
    navigation.goBack(); // Navigate back after saving
  } catch (error) {
    console.log("Error saving note:", error);
```

Figure 5.27 Code snippet of saving a new note to the database

5.9 View previous DASS-21 stress score value

One of the main functionalities of the proposed application is to allow user to take part in the DASS-21 questionnaire to allow themselves to understand their initial mental score value before conducting any stress reducing activities. Users are able to constantly take part in this questionnaire after conducting the activity provided in the application to monitor their stress score value. Therefore, the system is designed to store all the previous stress score value in the database as shown in the Figure 5.28. Similar to the above-mentioned functions, all these stress score value will be stored into the database linked under the user account. When the user reach the profile page of the application, the system will fetch the data according to the current active user ID from the database and display the stress score where the latest stress score will be displayed first. The stress score displayed will consists of the score and the uploaded timestamp. By doing that, user can always check back the stress score they have obtained ever since they have started using the application. The stress score displayed in number will be differentiated with colour in order to identify the stress score range. For example, red colour of stress score represents the user had experienced heavy stress during that time.

Hence, they can monitor stress score on different time and perhaps investigate or retrace back on why they have such certain scores. If they have higher score value, this indicates that they are under heavy stress during that time. Moreover, users can also check back their notes documented earlier to reflect back the events happened.



15:55 | 0.4KB/s

Profile

CH

Joined 249 days ago

View your previous stress scores

12/21 25/07/2023 11:13:34 PM

16/21 25/07/2023 11:13:15 PM

My Profile

Setting

Meet HepiPal, your adorable stress-busting companion!

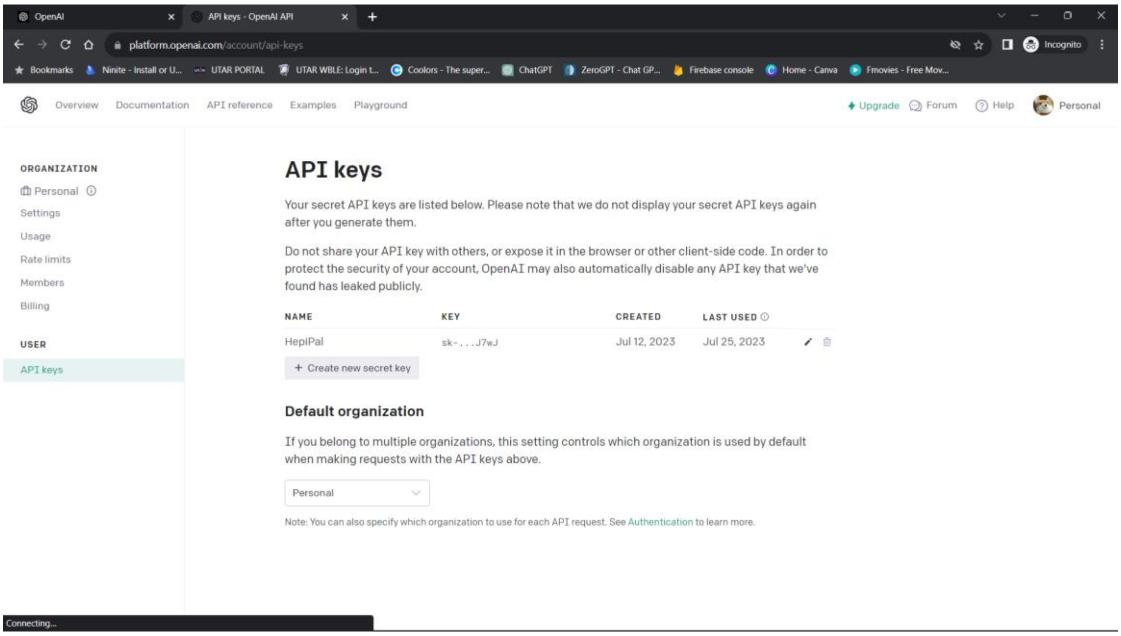
Take me there

Log Out

Figure 5.28 Past stress score values are displayed for user reference

5.10 Integrating a mini-AI chatbot - HepiPal

The proposed application is designed to help user to understand more about their own feelings inside when they are under stress, especially when they have the negative feelings from different factors such as academic stress, family issue, financial burden, and relationship issues that contribute to stress. When individuals find themselves with limited options for conversation, they can turn to the integrated mini-AI chatbot for engaging discussions, allowing them to converse with the chatbot and explore various topics including health and wellness such as mental health. For instance, users can interact with the chatbot to further understand the cause of their mental issues and how to mitigate them. Therefore, the mini chatbot is developed by using deep learning attention-based model, which is a Generative Pre-trained Transformer (GPT) model. GPT model is currently the state of art architecture because the model can process the whole input sequence, not word by word and this process can be parallelised. This makes the whole process to be faster and accurate because it saves a lot of computational time and cost [31]. By implementing GPT model inside the proposed application, the application can perform various tasks such as create conversational agents, answer questions about a knowledge base, and analyse text [32]. These tasks are performed by the mini chatbot application as an agent to initiate the conversation. In order, o implement the GPT model into the application and to develop a chatbot, Application Programming Interface (API) services from OpenAI is integrated to access the GPT model, which the GPT 3.5 model. It is one of the latest and powerful models to understand and generate natural language or code [33]. The choice of the text-davinci-003 model from GPT 3.5 is based on its superior ability to perform various language tasks with higher quality, generate longer outputs, and exhibit more consistent adherence to instructions compared to the previous, outdated model [32]. Before implementing the GPT model inside, an OpenAI platform is created to generate and obtain a usable API key from the platform as shown in the Figure 5.29 below.



API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED
HepiPal	sk-...J7wJ	Jul 12, 2023	Jul 25, 2023

+ Create new secret key

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

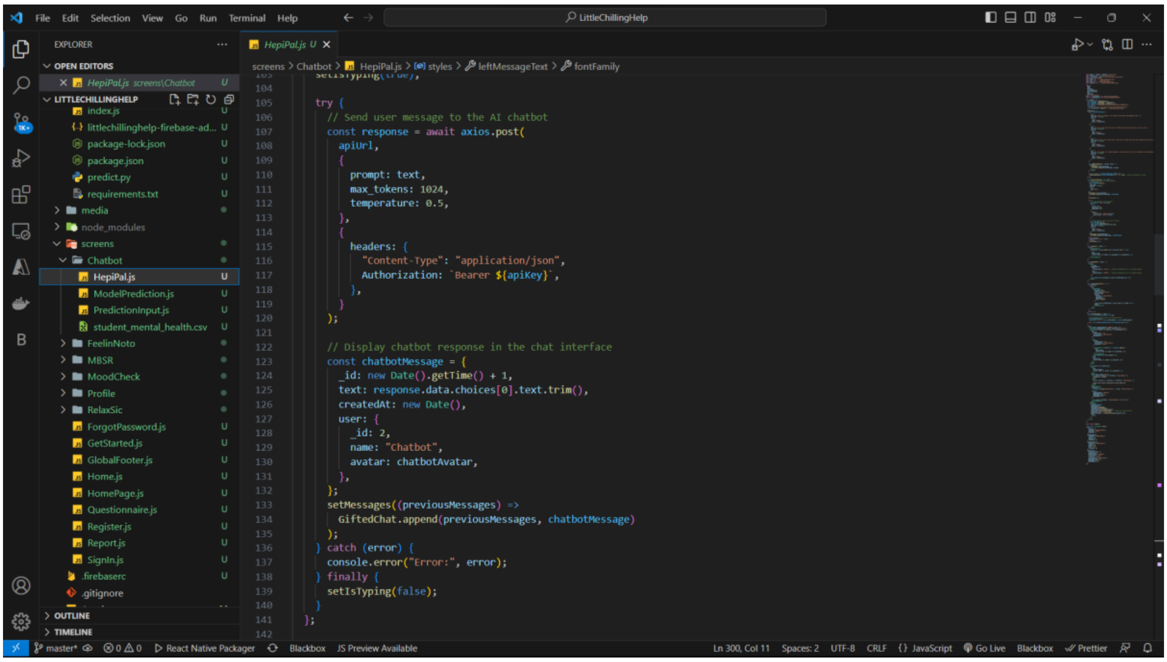
Personal

Note: You can also specify which organization to use for each API request. See Authentication to learn more.

Figure 5.29 Obtaining a usable API secret key from OpenAI for chatbot development

After obtaining the API key, the authorisation process is setup in the application code. This is because for every time users wish to use the chatbot function, it needs to go under authentication by checking the API key with the system. In order to guarantee the seamless exchange of messages with the chatbot, specific parameters within the

GPT model must be configured in accordance with the application's requirements to maintain the desired text quality in the chatbot's responses.



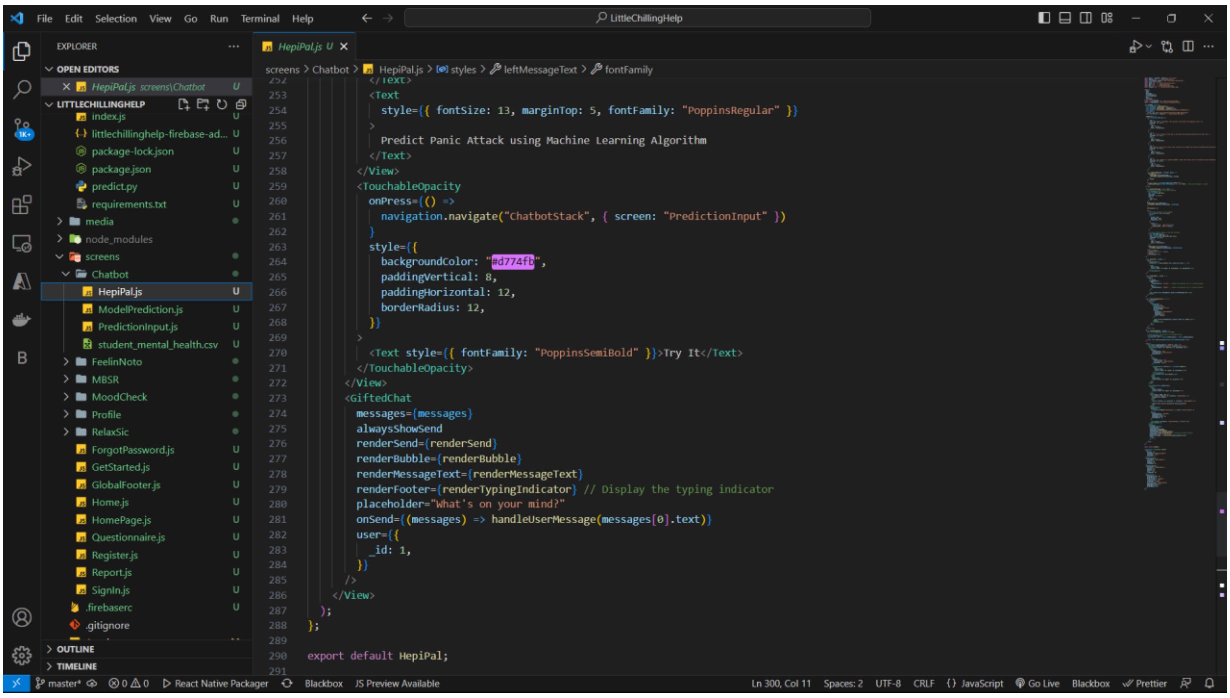
LittleChillingHelp
HepiPal.js U
screens > Chatbot > HepiPal.js > styles > leftMessageText > fontFamily
EXPLORER
OPEN EDITORS
LITTLECHILLINGHELP
OUTLINE
TIMELINE
master*
Ln 300, Col 11 Spaces: 2 UTF-8 CRLF JavaScript Go Live Blackbox Prettier

```javascript
  setIsTyping(true);

  try {
    // Send user message to the AI chatbot
    const response = await axios.post(
      apiUrl,
      {
        prompt: text,
        max_tokens: 1024,
        temperature: 0.5,
      },
      {
        headers: {
          "Content-Type": "application/json",
          Authorization: `Bearer ${apiKey}`,
        },
      }
    );

    // Display chatbot response in the chat interface
    const chatbotMessage = {
      _id: new Date().getTime() + 1,
      text: response.data.choices[0].text.trim(),
      createdAt: new Date(),
      user: {
        _id: 2,
        name: "Chatbot",
        avatar: chatbotAvatar,
      },
    };
    setMessages((previousMessages) =>
      GiftedChat.append(previousMessages, chatbotMessage)
    );
  } catch (error) {
    console.error("Error:", error);
  } finally {
    setIsTyping(false);
  }
};
```

Figure 5.30 Code snippet of chatbot parameters configuration

Based on the Figure 5.30 above, there are three parameters to be setup every time the system passes the text messages from the user to the model. The first parameter is the "prompt". In the proposed application, it contains a chatbot feature, where the prompt is the text message from the user during the whole ongoing conversation. The second parameter is the "max_tokens", which is read and written by the language model in chunks of the text. The total number of tokens in an API call affects the cost of the API call, and the time needed for the chatbot to generate the output message, as writing more tokens consumes more time [32]. Since the chatbot application primarily serves as a conversational agent, it occasionally offers advice or suggestions to the users. Hence, it should avoid generating lengthy tokens or text, considering that users may already be mentally fatigued and prefer concise responses. Therefore, the "max_tokens" of the chatbot parameter is set to 1024. The earlier mentioned requirements for this chatbot emphasize the need for the third parameter, "temperature", to consistently generate relevant output messages, thereby contributing to assisting users in managing their stress. The temperature parameter is set to be 0.5 because lower value generates more consistent result and higher value generate more variability output [32]. Once the backend services are fully configured, the connection between the backend services and the GPT model is successfully established.

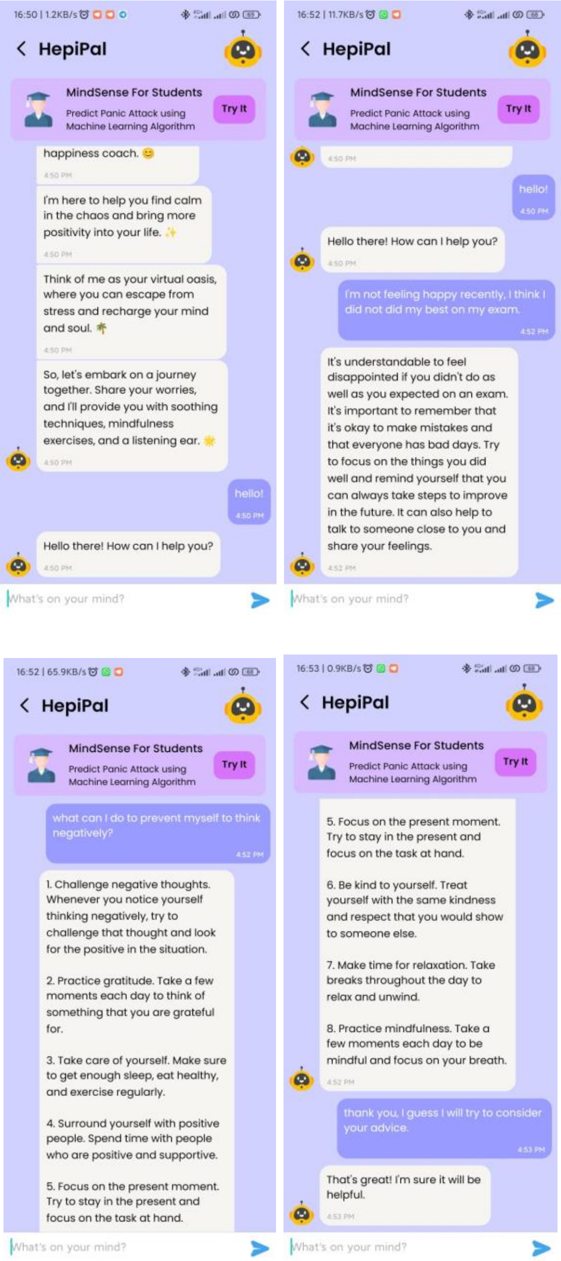
In terms of application interface development, the chatbot application is designed to resemble the style and user experience of a messaging application, ensuring that users find it familiar in both design and usage. As a result, the GiftedChat library is implemented to create the application interface. This library increases the capability to design an interface that resembles a messaging application, achieving the desired look and feel, as shown in the Figure 5.31 below. Every time when a message is written, new messages will be appended to the chat list provided by the function of GiftedChat library. By appending the latest message to the list, it can display the latest messages at the bottom of the chat interface.



```
          </Text>
          <Text
            style={{ fontSize: 13, marginTop: 5, fontFamily: "PoppinsRegular" }}
          >
            Predict Panic Attack using Machine Learning Algorithm
          </Text>
        </View>
        <TouchableOpacity
          onPress={() =>
            navigation.navigate("ChatbotStack", { screen: "PredictionInput" })
          }
          style={{
            backgroundColor: "#d774fb",
            paddingVertical: 8,
            paddingHorizontal: 12,
            borderRadius: 12,
          }}
        >
          <Text style={{ fontFamily: "PoppinsSemiBold" }}>Try It</Text>
        </TouchableOpacity>
      </View>
      <GiftedChat
        messages={messages}
        alwaysShowSend
        renderSend={renderSend}
        renderBubble={renderBubble}
        renderMessageText={renderMessageText}
        renderFooter={renderTypingIndicator} // Display the typing indicator
        placeholder="What's on your mind?"
        onSend={(messages) => handleUserMessage(messages[0].text)}
        user={{
          _id: 1,
        }}
      />
    </View>
  );
};

export default HepiPal;
```

Figure 5.31 Implementation of GiftedChat library in the chatbot interface

There are only two types of messages, the user message and chatbot message. Both message types include common attributes such as the message ID, typically representing the timestamp when the message is generated, the message text, message creation date, as well as the user's ID, name, and avatar. These attributes are stored in the library, and certain ones, especially the message generation date and time, are displayed in the chat bubble to provide timestamp information for the message. Lastly, before the user engage the communication and pass the input message to the backend language model, default welcoming messages is generated. The welcoming message is presented according to the figure 5.32. Prior to the user entering their initial input message, the system will automatically display the predefined welcome messages that were programmed into the application. These messages serve to welcome the user to the mini-AI chatbot application and provide a brief explanation of the chatbot's capabilities.



happiness coach. 😊
4:50 PM

I'm here to help you find calm in the chaos and bring more positivity into your life. ✨
4:50 PM

Think of me as your virtual oasis, where you can escape from stress and recharge your mind and soul. 🌴
4:50 PM

So, let's embark on a journey together. Share your worries, and I'll provide you with soothing techniques, mindfulness exercises, and a listening ear. 🌟
4:50 PM

hello!
4:50 PM

Hello there! How can I help you?
4:50 PM

4:50 PM

hello!
4:50 PM

Hello there! How can I help you?
4:50 PM

I'm not feeling happy recently, I think I did not did my best on my exam.
4:52 PM

It's understandable to feel disappointed if you didn't do as well as you expected on an exam. It's important to remember that it's okay to make mistakes and that everyone has bad days. Try to focus on the things you did well and remind yourself that you can always take steps to improve in the future. It can also help to talk to someone close to you and share your feelings.
4:52 PM

what can I do to prevent myself to think negatively?
4:52 PM

1. Challenge negative thoughts. Whenever you notice yourself thinking negatively, try to challenge that thought and look for the positive in the situation.

2. Practice gratitude. Take a few moments each day to think of something that you are grateful for.

3. Take care of yourself. Make sure to get enough sleep, eat healthy, and exercise regularly.

4. Surround yourself with positive people. Spend time with people who are positive and supportive.

5. Focus on the present moment. Try to stay in the present and focus on the task at hand.

5. Focus on the present moment. Try to stay in the present and focus on the task at hand.

6. Be kind to yourself. Treat yourself with the same kindness and respect that you would show to someone else.

7. Make time for relaxation. Take breaks throughout the day to relax and unwind.

8. Practice mindfulness. Take a few moments each day to be mindful and focus on your breath.
4:52 PM

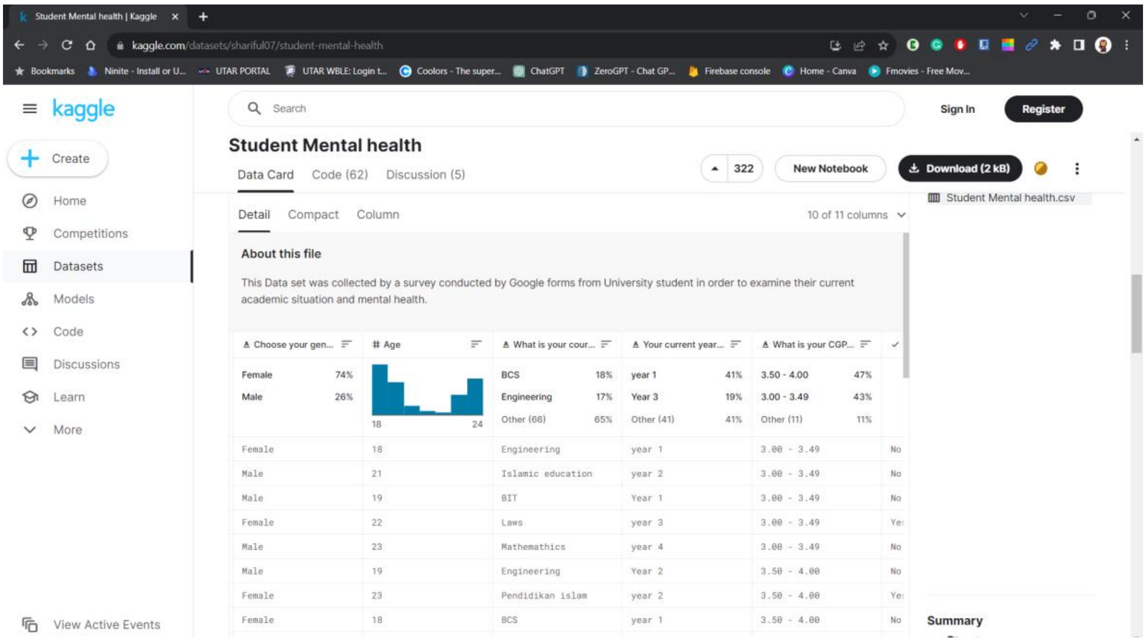
thank you, I guess I will try to consider your advice.
4:53 PM

That's great! I'm sure it will be helpful.
4:53 PM

Figure 5.32 Demonstration of the chatbot's responses towards user concerns

5.11 Creating a Panic Attack Prediction Tool in Python

In the proposed application, a panic attack prediction tool is developed. The functions primarily target aims at young adults to help them assess the likelihood of experiencing a panic attack due to accumulated stress. In order to develop such functions, the model implements the K-Nearest Neighbours (KNN) classification algorithm. The predictive capabilities of this model rely on training data collected from students worldwide, specifically academic qualification data gathered via Google Forms as shown in the Figure 5.33 below. This dataset is the result of a statistical research initiative that examines the impact of mental health on students' academic performance, particularly their Cumulative Grade Point Average (CGPA).



Student Mental health

About this file

This Data set was collected by a survey conducted by Google forms from University student in order to examine their current academic situation and mental health.

Choose your gen...	Age	What is your cour...	Your current year...	What is your CGP...	
Female 74% Male 26%		BCS 18% Engineering 17% Other (66) 65%	year 1 41% Year 3 19% Other (41) 41%	3.50 - 4.00 47% 3.00 - 3.49 43% Other (11) 11%	
Female	18	Engineering	year 1	3.00 - 3.49	No
Male	21	Islamic education	year 2	3.00 - 3.49	No
Male	19	BIT	Year 1	3.00 - 3.49	No
Female	22	Laws	year 3	3.00 - 3.49	Ye
Male	23	Mathemathics	year 4	3.00 - 3.49	No
Male	19	Engineering	Year 2	3.50 - 4.00	No
Female	23	Pendidikan islam	year 2	3.50 - 4.00	Ye
Female	18	BCS	year 1	3.50 - 4.00	No

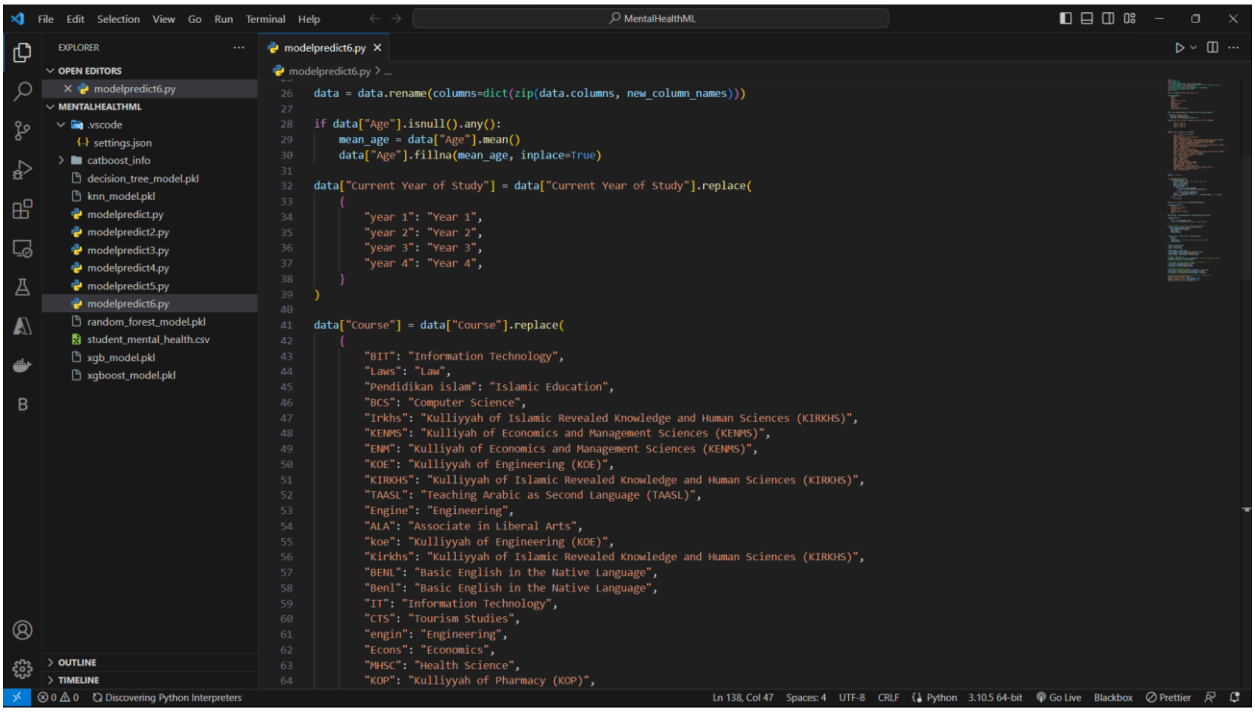
Figure 5.33 Student mental health dataset on Kaggle

Before constructing the model, the dataset undergoes a preprocessing phase. It's worth noting that this dataset is relatively small, containing only 100 records, and is obtained from Kaggle. As to ensure consistency and accuracy in the results, the dataset is subjected to preprocessing and cleaning during the development stage. This is a crucial step in preventing inconsistencies that could affect accuracy. The dataset comprises a total of 10 data attributes, with 7 being used as input variables (x variable) and the panic attack attribute serving as the target variable (y variable). The dataset comprises information on academic qualifications, age, gender, current year of study, course of study, CGPA range, marital status, seek of special treatment, and the target variable, panic attack. The Table 5.1 below provides detailed descriptions for each data attribute.

Table 5.1 Data attributes of the student mental health dataset

Data attribute	Variable Type	Example	Remark
Age	Numerical variable	25	The age value must be expressed as a whole integer.
Gender	Categorical variable	Female	The gender value should only permit 'Male' or 'Female' as valid entries.
Course studying	Categorical variable	Computer Science	The course of study should allow for the full name of the course, rather than requiring only the use of abbreviated versions.
Current year of study	Categorical variable	Year 3	The accepted values range from "Year 1" to "Year 4" only.
CGPA range	Categorical variable	2.00 - 2.49	This value is indicated by a range between 0 and 4.00, rather than a specific CGPA value.
Marital Status	Categorical variable	Yes	A "Yes" indicates a married status, while "No" signifies a single status.
Seek of special treatment	Categorical variable	Yes	This attribute indicates whether the user has previously sought special treatment, such as consulting a mental health professional or counsellor.
Panic Attack	Categorical variable (Target variable)	Yes	This value signifies that the user has previously experienced a panic attack in relation to their current academic background.

Text formatting is applied to maintain data consistency. Null values are identified in the "age" variable and replaced with the mean age value from the dataset. Text formatting is also performed on attributes like “studying year” and “course of study” to ensure uniformity. Once the dataset is thoroughly cleaned, it is split into training, testing, and validation sets.



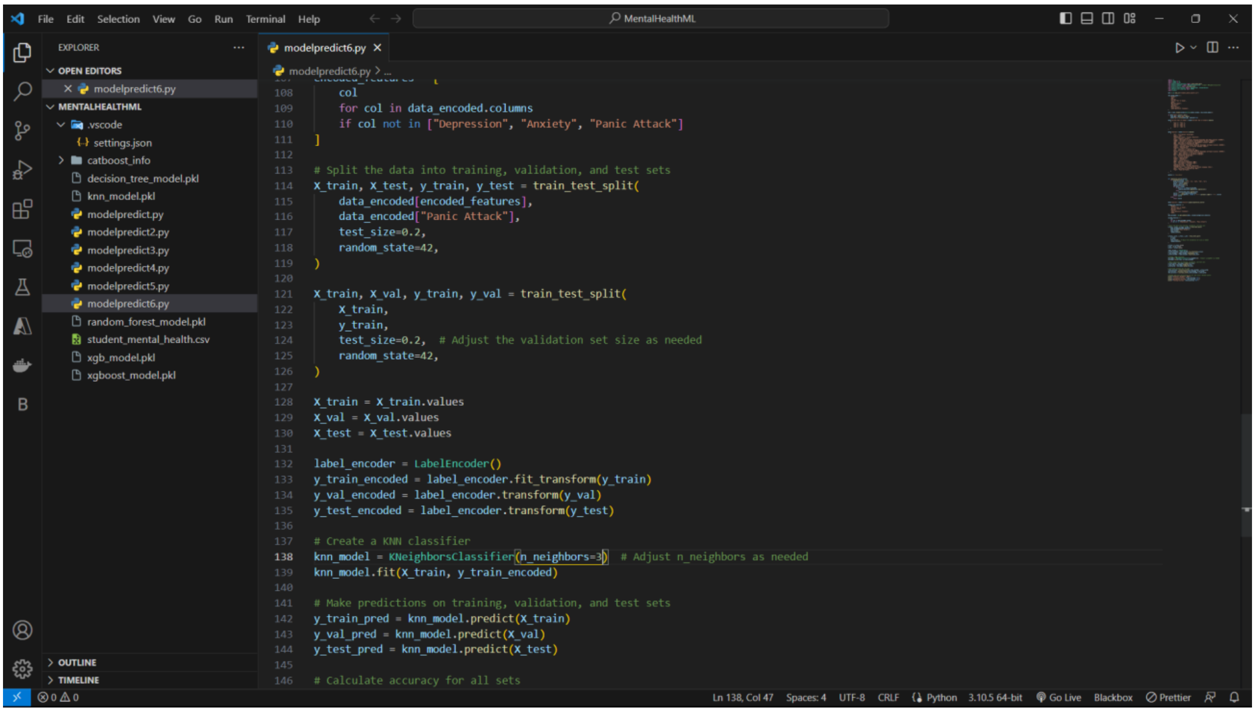
```
data = data.rename(columns=dict(zip(data.columns, new_column_names)))

if data["Age"].isnull().any():
    mean_age = data["Age"].mean()
    data["Age"].fillna(mean_age, inplace=True)

data["Current Year of Study"] = data["Current Year of Study"].replace(
    {
        "year 1": "Year 1",
        "year 2": "Year 2",
        "year 3": "Year 3",
        "year 4": "Year 4",
    }
)

data["Course"] = data["Course"].replace(
    {
        "BIT": "Information Technology",
        "Laws": "Law",
        "Pendidikan islam": "Islamic Education",
        "BCS": "Computer Science",
        "Irkhs": "Kulliyyah of Islamic Revealed Knowledge and Human Sciences (KIRKHS)",
        "KENMS": "Kulliyah of Economics and Management Sciences (KENMS)",
        "ENM": "Kulliyah of Economics and Management Sciences (KENMS)",
        "KOE": "Kulliyyah of Engineering (KOE)",
        "KIRKHS": "Kulliyyah of Islamic Revealed Knowledge and Human Sciences (KIRKHS)",
        "TAASL": "Teaching Arabic as Second Language (TAASL)",
        "Engine": "Engineering",
        "ALA": "Associate in Liberal Arts",
        "koe": "Kulliyyah of Engineering (KOE)",
        "Kirkhs": "Kulliyyah of Islamic Revealed Knowledge and Human Sciences (KIRKHS)",
        "BENL": "Basic English in the Native Language",
        "Benl": "Basic English in the Native Language",
        "IT": "Information Technology",
        "CTS": "Tourism Studies",
        "engin": "Engineering",
        "Econs": "Economics",
        "MHSC": "Health Science",
        "KOP": "Kulliyyah of Pharmacy (KOP)",
```

Figure 5.34 Code snippets of text formatting during data preprocessing stage

In order to prevent overfitting and enhance model performance, a portion of the training set is reserved for the validation set. This validation set plays a crucial role in evaluating the model’s performance and facilitating hyperparameter tuning. After dataset splitting is complete, the model is ready for training.

Several machine learning classification algorithms, including K-Nearest Neighbours (KNN) [35], XGBoost [36], Random Forest [37], and Decision Tree [38], are employed to train the model. After evaluating the results of multiple training rounds, KNN is selected as the most suitable algorithm due to its higher accuracy compared to the others. The performance evaluation of the models will be explained in Chapter 6. KNN is particularly well-suited for smaller datasets with minimal noise and labelled data. It also demonstrates faster execution times for smaller datasets.

The primary parameter to configure in the KNN classifier is the value of K, representing the number of neighbours considered in the algorithm. Based on the Figure 5.35 below, given the dataset’s smaller size, lack of noise, and minimal outliers, a K value of 5 is chosen. Predictions are made based on the majority class among the nearest neighbours.



```python
        col
        for col in data_encoded.columns
        if col not in ["Depression", "Anxiety", "Panic Attack"]
    ]

    # Split the data into training, validation, and test sets
    X_train, X_test, y_train, y_test = train_test_split(
        data_encoded[encoded_features],
        data_encoded["Panic Attack"],
        test_size=0.2,
        random_state=42,
    )

    X_train, X_val, y_train, y_val = train_test_split(
        X_train,
        y_train,
        test_size=0.2,  # Adjust the validation set size as needed
        random_state=42,
    )

    X_train = X_train.values
    X_val = X_val.values
    X_test = X_test.values

    label_encoder = LabelEncoder()
    y_train_encoded = label_encoder.fit_transform(y_train)
    y_val_encoded = label_encoder.transform(y_val)
    y_test_encoded = label_encoder.transform(y_test)

    # Create a KNN classifier
    knn_model = KNeighborsClassifier(n_neighbors=3)  # Adjust n_neighbors as needed
    knn_model.fit(X_train, y_train_encoded)

    # Make predictions on training, validation, and test sets
    y_train_pred = knn_model.predict(X_train)
    y_val_pred = knn_model.predict(X_val)
    y_test_pred = knn_model.predict(X_test)

    # Calculate accuracy for all sets
```

Figure 5.35 Configuration of parameters of KNN classifier during training

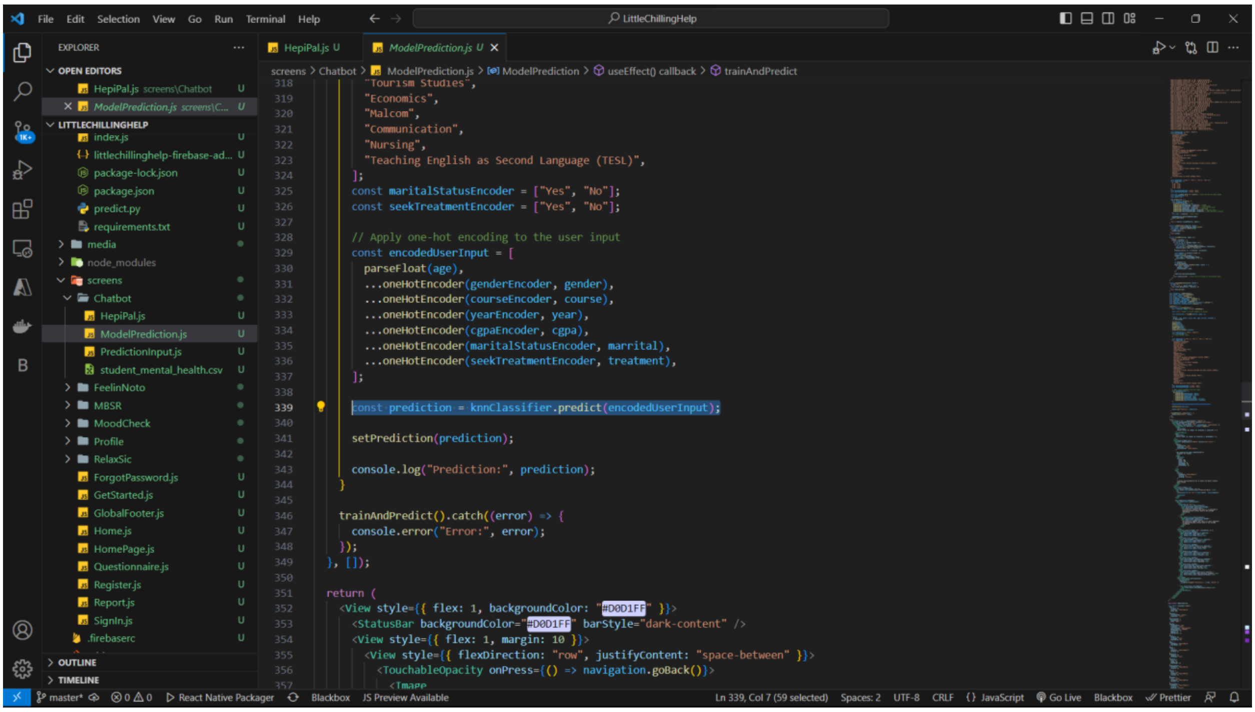
Upon completing the model training process, the model is saved as a pkl file in the Python module, making it accessible for use in the React Native application.

5.12 Creating a Panic Attack Prediction Tool and Implementing the KNN Model in React Native

Following the generation of the prediction model from the Python module, the next step involves integrating this model into the React Native application to create a panic attack prediction tool. The model is saved as a pkl file, a format created by the Python module “pickle” for serializing objects to disk and deserializing them at runtime [34]. This file will be an essential component of the machine learning classification utilized in the prediction tool.

In order to implement the KNN model in the React Native application, the pkl model file must be imported into the JavaScript-based application code. The pkl file serves as the machine learning model's backbone, allowing it to make predictions based on user input.

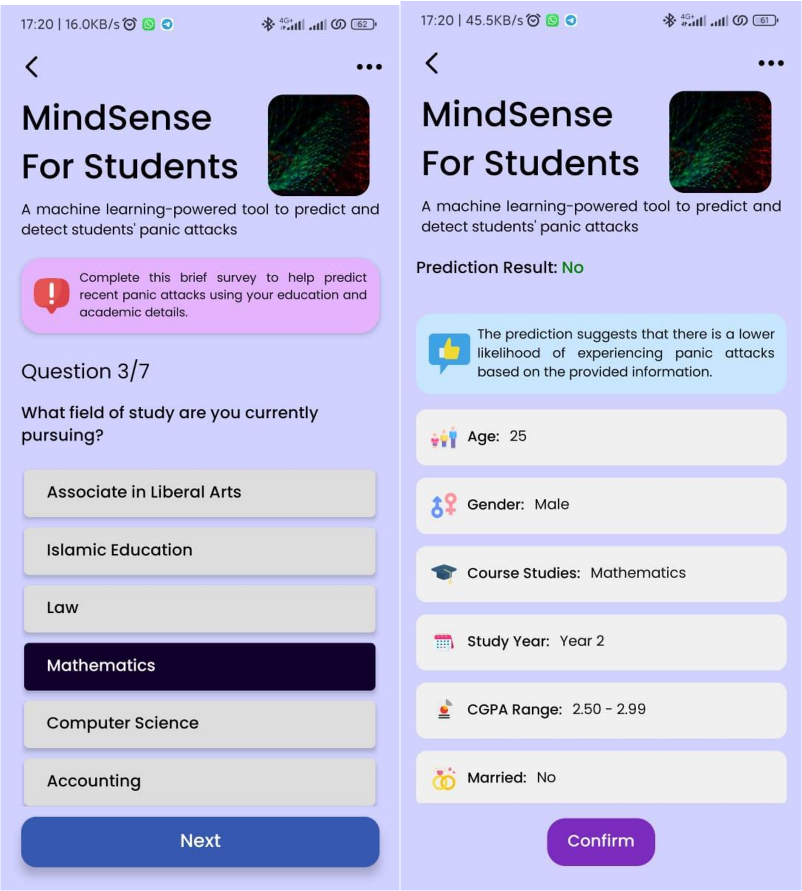
In the React Native interface of the prediction tool, seven input spaces are provided for users to input relevant information, especially pertaining to their academic background. These input questions align with the dataset attributes used as input variables for predicting outcomes. As to streamline the user experience, options lists are included for some of the questions, enabling users to select responses, conveniently. These option lists also help prevent input errors that might hinder the model's ability to make accurate predictions. Once users have completed the input questions, their responses are sent to the backend program, which incorporates the previously trained model. The input data is processed by the backend program, preparing it for prediction. Subsequently, the processed inputs are passed to the trained model through a JavaScript function called "knnClassifier". Within the "knnClassifier" function, the K-Nearest Neighbours (KNN) classifier is executed, leveraging the trained model to predict potential outcomes. The KNN classifier utilizes the same parameter settings as during training, with a value of k set to 3 to maintain the optimal accuracy achieved during training as shown in the Figure 5.36 below.


HepiPal.js U
ModelPrediction.js U
screens > Chatbot > ModelPrediction.js > ModelPrediction > useEffect() callback > trainAndPredict
LittleChillingHelp
Ln 339, Col 7 (59 selected)

```
        "Tourism Studies",
        "Economics",
        "Malcom",
        "Communication",
        "Nursing",
        "Teaching English as Second Language (TESL)",
      ];
      const maritalStatusEncoder = ["Yes", "No"];
      const seekTreatmentEncoder = ["Yes", "No"];

      // Apply one-hot encoding to the user input
      const encodedUserInput = [
        parseFloat(age),
        ...oneHotEncoder(genderEncoder, gender),
        ...oneHotEncoder(courseEncoder, course),
        ...oneHotEncoder(yearEncoder, year),
        ...oneHotEncoder(cgpaEncoder, cgpa),
        ...oneHotEncoder(maritalStatusEncoder, marrital),
        ...oneHotEncoder(seekTreatmentEncoder, treatment),
      ];

      const prediction = knnClassifier.predict(encodedUserInput);

      setPrediction(prediction);

      console.log("Prediction:", prediction);
    }

    trainAndPredict().catch((error) => {
      console.error("Error:", error);
    });
  }, []);

  return (
    <View style={{ flex: 1, backgroundColor: "#D0D1FF" }}>
      <StatusBar backgroundColor="#D0D1FF" barStyle="dark-content" />
      <View style={{ flex: 1, margin: 10 }}>
        <View style={{ flexDirection: "row", justifyContent: "space-between" }}>
          <TouchableOpacity onPress={() => navigation.goBack()}>
            <Image
```

Figure 5.36 Code snippet of predicting outcome using the KNN trained model from Python

During the model execution, it generates an output to indicate whether the user is likely to experience a panic attack based on their current academic background information especially the current course studying and the CGPA result. Based on the Figure 5.37, the output is then displayed to the user, typically as a “Yes” or “No” result, accompanied by a brief comment explaining the prediction outcome.



17:20 | 16.0KB/s

MindSense For Students

A machine learning-powered tool to predict and detect students' panic attacks

Complete this brief survey to help predict recent panic attacks using your education and academic details.

Question 3/7

What field of study are you currently pursuing?

- Associate in Liberal Arts
- Islamic Education
- Law
- Mathematics
- Computer Science
- Accounting

Next

17:20 | 45.5KB/s

MindSense For Students

A machine learning-powered tool to predict and detect students' panic attacks

Prediction Result: No

The prediction suggests that there is a lower likelihood of experiencing panic attacks based on the provided information.

Age: 25

Gender: Male

Course Studies: Mathematics

Study Year: Year 2

CGPA Range: 2.50 - 2.99

Married: No

Confirm

Figure 5.37 Panic attack prediction tool interface

This prediction tool empowers users to mentally prepare themselves or monitor their stress levels more closely, particularly if they are experiencing severe stress. By offering insights into the likelihood of a panic attack, users can take proactive steps to manage their stress levels, potentially engaging in activities like Mindfulness-Based Stress Reduction (MBSR) to alleviate stress and restore their mental well-being.

CHAPTER 6: System Evaluation and Discussion

6.1 Comparing overall performance between models

The panic attack prediction tool is constructed using a binary classification model to forecast whether users will experience panic attacks due to stress conditions. The panic attack prediction tool has been specifically tailored for users within the student community. This tool is intended to forecast the probability of users experiencing panic attacks in the future, based on input data related to their current academic background, including factors such as the course studied and the examination result. In order to construct the classification model before its integration into the proposed application, a suitable dataset has been selected. This dataset comprised of one hundred rows of records collected from students worldwide with the data attributes including age, gender, course of study, current year of study, current CGPA range, marital status, and seeking of special treatment as mentioned in Chapter 5 Section 5.11. Four classifiers were chosen for training and their results were assessed to identify the best-performing model for integration into the proposed application. The four evaluated classifiers are XGBoost [36], Decision Tree [38], Random Forest [37], and K-Nearest Neighbours (KNN) [35].

6.1.1 XGBoost

XGBoost operates by iteratively building an ensemble of decision trees, where each tree contributes to the final classification by addressing the errors made by its predecessors. This process begins with an initial tree and continues sequentially, with each subsequent tree aiming to improve the classification model's accuracy. XGBoost employs a weighted sum of the individual tree predictions to make the final classification decision. The model assigns different weights to each tree's prediction based on their respective performance in minimizing classification errors during training [36].

As to further enhance its classification capabilities, XGBoost employs several hyperparameters. For instance, the "learning rate" controls the step size at which the model updates its predictions with each new tree. A smaller learning rate makes the process more conservative, while a larger one allows for faster convergence but may be prone to overfitting [41]. Additionally, the "max depth" hyperparameter sets an upper limit on the depth of each decision tree, preventing them from becoming overly complex and overfitting the training data.

6.1.2 Decision Tree

Decision Tree is a straightforward and interpretable machine learning model, operates by recursively partitioning data into subsets based on the most influential features, creating a tree-like structure for decision-making [38]. The core principle behind Decision Trees is to select the feature that best splits the data at each node, optimizing a criterion such as Gini impurity or information gain. This chosen feature becomes the node's decision rule, and the data is divided into child nodes accordingly. This process repeats recursively for each child node, creating a hierarchical structure. Ultimately, the leaf nodes of the tree represent the model's final decisions or classifications, in which it predicts the likelihood of getting a panic attack.

In Decision Trees, the "max depth" hyperparameter controls the depth or complexity of the tree. A smaller value limits the tree's depth, preventing overfitting, while a larger value allows for a more detailed tree. The "min samples split" hyperparameter determines the minimum number of data points required to split a node. A smaller value leads to finer splits, while a larger value results in fewer, broader splits, helping control model complexity [42].

6.1.3 Random Forest

Random Forest is an ensemble learning algorithm that excels at classifying data by creating a collection of decision trees. It constructs a multitude of decision trees simultaneously, with each tree being trained on a random subset of the data and a subset of the features [37]. This randomness injected into the training process helps prevent overfitting and encourages diversity among the trees.

In order to classify data, Random Forest aggregates the predictions of all its individual trees. Each tree "votes" for a class, and the class with the most votes becomes the final classification decision. This ensemble approach increases the model's robustness and generalization capabilities, making it effective at handling various data types and complex relationships [37].

Two crucial hyperparameters in Random Forest are "n estimators", which determines the number of trees in the forest, and "max features", which specifies the maximum number of features to consider when splitting a node. Tuning these hyperparameters is essential to strike the right balance between model complexity and accuracy. A higher number of trees generally leads to better performance, but it also increases computational cost, while adjusting “max features” can help control overfitting [43].

6.1.4 K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a versatile classification algorithm that relies on non-parametric and instance-based principles. It stands out for its ability to recognise data patterns at a local level, making it particularly suited for situations where data clusters together [35]. KNN classifies data points by evaluating their proximity to neighbouring data points within the feature space.

In order to classify a new data point, KNN calculates the distances between this point and its closest neighbours, typically using a chosen distance metric, such as Euclidean distance. The number of neighbours considered in this process is determined by the hyperparameter known as “K”. For instance, when K is set to 5, KNN identifies the five nearest data points to the new data point. It then examines the class labels of these nearest neighbours and assigns the majority class label to the new data point [44].

The choice of the K value plays a significant role in KNN’s performance. A smaller K value results in a more localized decision-making process, focusing on the nearest neighbours and their inherent data patterns [44]. This can make KNN highly sensitive to noise or outliers. In contrast, a larger K value broadens the scope of influence, leading to more stable but potentially less precise predictions.

6.1.5 Performance Evaluation

All the four mentioned classifiers were employed to train, validate, and test the dataset. The performance of these classifiers was evaluated using accuracy evaluation metrics as shown in Equation 6.1 below.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \qquad \text{e.q (6.1)}$$

Accuracy is a commonly used performance metric in machine learning, especially for classification problems. It measures the proportion of correctly predicted samples out of the total number of samples in the dataset. Based on the Equation 6.1 above, before obtaining the accuracy values from the prediction, True Positives (TP), True Negatives

(TN), False Positives (FP), and False Negatives (FN) has to be determined. Then, the accuracy should be calculated by dividing the sum of TP and TN by the total number of samples in the dataset.

The accuracy values for the mentioned four classifiers were measured for training, validation, and testing of the dataset. These accuracy values obtained for each classifier are subsequently analysed and compared as illustrated in Table 6.1 below.

Table 6.1 Accuracy values between the models

Classifier	Hyperparameter Configuration	Training Accuracy	Validation Accuracy	Testing Accuracy
Random Forest	n estimator = 50 max feature = “sqrt”	0.9895	0.6875	0.5841
Decision Tree	max depth = 5 min sample split = 5	0.9713	0.5625	0.5714
XGBoost	learning rate = 0.01 max depth = 4	0.9531	0.6072	0.6190
KNN	k = 5	0.7344	0.6182	0.6667

From Table 6.1, Random Forest achieved the highest training accuracy of 0.9895. This shows that the model has learned to fit the training data well because it was able to correctly predict the labels for 98.95% of the training samples. It also obtained the validation accuracy of 0.6875, indicating overfitting to the training data. The drop in accuracy from training to validation indicates that the model may have started to capture noise or specific details of the training data that don’t generalize well to other data. Besides that, the testing accuracy of 0.5841 were comparatively lower, suggesting that the model has performed worse in which it has not learned the underlying patterns in the data but has instead memorized the training data.

Decision Tree exhibited an almost perfect training accuracy of 0.9713, which raised concerns about overfitting, especially considering its validation accuracy of 0.5625. The accuracy dropped almost half between the training and validation accuracy. The model delivered similar result as the Random Forest model, as it learned to fit the data extremely well. The testing accuracy is only marginally better at 0.5714, indicating that the model's overfitting issue persists when it is tested on completely unseen data. Overfitting in a Decision Tree occurs when the tree is grown too deep and becomes overly complex. In this case, the tree has become so complex in early stage that it has essentially memorized the training data, including its noise and outliers, rather than learning the underlying patterns.

On the other hand, XGBoost exhibited a notable training accuracy of 0.9531, correctly predicting labels for 95.31% of the training samples. However, this high training accuracy raised concerns about potential overfitting. While its validation accuracy of 0.6072 outperformed that of Random Forest and Decision Tree, it did not reach exceptionally high levels. A significant difference between training and validation accuracy often signals overfitting, indicating that the model is not effectively generalizing to unseen data. XGBoost also achieved a testing accuracy of 0.6190, representing an improvement compared to Random Forest and Decision Tree models. However, it fell short of achieving a very high testing accuracy. This suggests that while XGBoost may have improved its generalization ability compared to the first two models, it still encounters challenges in making accurate predictions on data it hasn't encountered during training.

Lastly, KNN achieved a training accuracy of 0.7344, which is a moderate accuracy level, and it is lower than the training accuracies of Random Forest, Decision Tree, and XGBoost. This indicates that KNN has reasonably captured patterns in the training data but is less prone to overfitting compared to the other models. Moreover, KNN obtained a validation accuracy of 0.6182. The drop in accuracy from training to validation is less severe than in some of the other models, suggesting that KNN may be more robust against overfitting. Notably, KNN demonstrated the highest testing accuracy of 0.6667 among the four models. This is a positive signal, indicating that KNN generalized well to unseen data. The balanced result of KNN between bias and variance suggests that KNN performed well without significant overfitting issues.

In summary, Random Forest, Decision Tree, and XGBoost exhibit overfitting tendencies due to their high training accuracy but lower performance on validation and testing datasets. Only KNN shows a balanced performance across all three sets. Therefore, KNN is selected as the model for developing the panic attack prediction tool because it strikes a balance between bias and variance, a crucial consideration for classification tasks, particularly when dealing with imbalanced datasets. Moreover, KNN is a simple and interpretable model, making it easier to work with. Its instance-based learning also benefits from faster performance due to the smaller dataset size, as the algorithm computes distances between data points more efficiently in lower-dimensional spaces [35]. However, further model tuning and feature engineering are necessary to enhance overall performance.

6.2 Hyperparameter Tuning for KNN model

The KNN classifier has been selected as the model to be utilized in constructing the panic attack prediction tool within the proposed application, owing to its advantages over other classifiers. However, to further enhance the model's overall performance and ensure accurate predictions when used by real-life users, hyperparameter tuning is required.

One challenge addressed is the relatively small dataset size used for model training. In order to mitigate this issue, Synthetic Minority Over-sampling Technique (SMOTE) has been employed for random oversampling. SMOTE generates synthetic samples for the minority class, effectively balancing both classes in the target variable within the dataset [39].

Additionally, the model's K value has been thoroughly examined by evaluating different K values during training. Various K values, ranging from 3 to 7, have been employed to train the KNN classifier model, and the resulting accuracy measurements have been collected and compared. The selection of the optimal K value requires careful consideration. Smaller K values can introduce noise into the dataset and exert a higher influence on the results, while larger K values may lead to increased computational complexity [40].

The accuracy values for the different configuration settings are presented in the Table 6.2 below.

Table 6.2 Accuracy values for different configuration settings of KNN model

Configuration	Training Accuracy	Validation Accuracy	Testing Accuracy
SMOTE, k=3	0.8068	0.6849	0.7037
SMOTE, k=5	0.7727	0.5625	0.4762
SMOTE, k=7	0.7045	0.5000	0.5238
k=3	0.6562	0.6250	0.6667
k=7	0.6719	0.7500	0.7143

Based on the observed results from Table 6.2, the KNN classifier model exhibited performance enhancement when configured with SMOTE and a k value of 3. A marginal improvement was identified, as the new training, validation, and testing accuracy surpassed those of the previous model version which was implementing the K value of 5. Hence, this KNN classifier model with k value of 3 will be integrated into the proposed application. However, it’s worth noting that selecting a low value of K, such as 3, introduces the possibility of noise in the data, may lead to higher variance, and makes the model more sensitive to outliers or minor fluctuations in the data.

CHAPTER 7: Conclusion

7.1 Conclusion and Novelty

The project has successfully achieved its development goals by incorporating a comprehensive set of features into the proposed application aimed at helping users evaluate and manage their stress levels effectively. These features are designed to provide a user-friendly approach manage stress levels and are shown below:

1. DASS-21 Questionnaire: Users are required to participate in the DASS-21 questionnaire, allowing them to gain insights into their initial mental well-being, particularly regarding stress. This feature is especially valuable for new users seeking to assess their current emotional state.

2. MBSR Meditation Exercises: Drawing from the principles of the 8-week MBSR course, the application condenses and extract the essence of mindfulness-based stress reduction (MBSR) into three categories of meditation exercises: Beginner, Intermediate, and Advanced. These exercises serve as one of the approaches for users seeking stress relief.

3. Mood Check-In: Users can regularly check in on their mood every day, fostering self-awareness and promoting emotional balance.

4. Quick Destress Feature (Breathing Exercises): For moments of immediate stress, the application offers quick and accessible breathing exercises to help users regain their calmness in a short time.

5. MBSR Relaxing Music Player: Users have access to a wide selection of relaxing music, enhancing their mindfulness and relaxation experience.

6. Daily Note Taking Feature: This feature allows users to document their thoughts and emotions daily, providing a space for self-inspection and emotional processing.

7. Mini AI-Chatbot Application – HepiPal: The application incorporates artificial intelligence to provide users with a virtual companion, HepiPal, which engages in conversations with users, encouraging them to express their innermost thoughts and emotions. Not only that, HepiPal offers suggestions and advice to help users reduce their stress levels.

8. Panic Attack Prediction Tool – MindSense: Leveraging a KNN classifier model, the application predicts the likelihood of users experiencing panic attacks based on their academic background. Students who frequently experience high levels of stress from their academic commitments will find this feature useful against elevation in the stress levels.

In summary, the proposed application is designed to offer users comprehensive set of features for evaluating, monitoring, and managing their stress levels over time. It encourages users to engage with their emotions and thoughts, which are key components of mindfulness in MBSR practices. Additionally, the integration of artificial intelligence through HepiPal and the predictive capabilities of MindSense using KNN classifier further enhance the application's utility in promoting stress reduction and emotional well-being.

7.2 Recommendations

There are several recommendations that could enhance the functionality and user experience of this project. Firstly, expanding the range of meditation exercises available within the application would provide users with more diverse options. This allows users to explore different meditation experiences and sensations, preventing monotony and increasing engagement with the application. Moreover, improving the note-taking feature by implementing speech-to-text recognition would make it more versatile and user-friendly. This addition allows the users to record their daily thoughts and experiences through voice, a format that many individuals prefer over typing lengthy text entries. As to further enhance user engagement, considering implementation of a notification system that can send reminders even when the application is not in use. This feature can prompt users to regularly participate in the stress questionnaire or perform daily mood check-ins, ensuring they stay actively involved in managing their stress levels. In terms of the KNN model used for the panic attack prediction tool, there is also room for improvement in accuracy. This can be accomplished by fine-tuning the hyperparameters, implementing transfer learning methods, and acquiring a larger dataset with a longer history of cases. Hence, the application can improve its usability, intuitiveness, and efficiency in assisting users in managing their stress and well-being by implementing these suggestions.

REFERENCES

[1] S. Panicker and P. Gayathri, "A survey of machine learning techniques in physiology based mental stress detection systems", Biocybernetics and Biomedical Engineering, vol. 39, no. 2, pp. 444-469, 2019. Available: 10.1016/j.bbe.2019.01.004.

[2] "Depression Anxiety Stress Scale-21 (DASS21)", Addiction Research Center. [Online]. Available: https://arc.psych.wisc.edu/self-report/depression-anxiety-stress-scale-21-dass21/. [Accessed: 05- Aug- 2022].

[3] M. Le, T. Tran, S. Holton, H. Nguyen, R. Wolfe and J. Fisher, "Reliability, convergent validity and factor structure of the DASS-21 in a sample of Vietnamese adolescents", PLOS ONE, vol. 12, no. 7, p. e0180557, 2017. Available: 10.1371/journal.pone.0180557.

[4] H. SARIÇAM, "The Psychometric Properties of Turkish Version of Depression Anxiety Stress Scale-21 (DASS-21) in Community and Clinical Samples", Journal of Cognitive-Behavioral Psychotherapy and Research, p. 1, 2018. Available: 10.5455/jcbpr.274847.

[5] "Cronbach's alpha – a measure of the consistency strength", Bachelorprint.eu. [Online]. Available: https://www.bachelorprint.eu/research/cronbachs-alpha/#1589198206530-3ba131a7-d138. [Accessed: 05- Aug- 2022].

[6] The Star. (2018, August). 29% of Malaysians have mental problems due to stress, says Lam Thye. https://www.thestar.com.my/news/nation/2018/08/05/lee-lam-thyemore-msians-will-suffer-from-depression-by-2020-due-to-stress.

[7] Lerner, K. L., & Lerner, B. W. (Eds.). (2008). The GALE ENCYCLOPEDIA of SCIENCE (fourth). Library of Congress Cataloging In Publication To Data.

[8] W. Hwang and H. Jo, "Evaluation of the Effectiveness of Mobile App-Based Stress-Management Program: A Randomized Controlled Trial", International Journal of Environmental Research and Public Health, vol. 16, no. 21, p. 4270, 2019. Available: 10.3390/ijerph16214270.

[9] J. Zhao, B. Freeman and M. Li, “Can Mobile Phone Apps Influence People’s Health Behavior Change? An Evidence Review”, Journal of Medical Internet Research, vol. 18, no. 11, p. e287, 2016. Available: 10.2196/jmir.5692.

[10] H. Payne, C. Lister, J. West and J. Bernhardt, “Behavioral Functionality of Mobile Apps in Health Interventions: A Systematic Review of the Literature”, JMIR mHealth and uHealth, vol. 3, no. 1, p. e20, 2015. Available: 10.2196/mhealth.3335.

[11] R. Lee and M. Jung, “Evaluation of an mHealth App (DeStressify) on University Students’ Mental Health: Pilot Trial”, JMIR Mental Health, vol. 5, no. 1, p. e2, 2018. Available: 10.2196/mental.8324.

[12]"Meditation and Sleep Made Simple - Headspace”, Headspace.com. [Online]. Available: https://www.headspace.com/. [Accessed: 22- Aug- 2022].

[13]"Headspace Health”, Headspacehealth.com. [Online]. Available: https://www.headspacehealth.com/. [Accessed: 22- Aug- 2022].

[14]"Shine: a well-being application and App Store Best of 2020”, Netguru.com, 2020. [Online]. Available: https://www.netguru.com/featured/shine-awarded-well-being-app. [Accessed: 22- Aug- 2022].

[15]"Shine | Calm Anxiety & Stress”, Theshineapp.com. [Online]. Available: https://www.theshineapp.com/. [Accessed: 22- Aug- 2022].

[16]"Amaha - Your Mental Health Partner", Amaha. [Online]. Available: https://www.amahahealth.com/. [Accessed: 22- Aug- 2022].

[17]"Smiling Mind”, Smiling Mind. [Online]. Available: https://www.smilingmind.com.au/. [Accessed: 22- Aug- 2022].

[18] M. Mani, D. Kavanagh, L. Hides and S. Stoyanov, “Review and Evaluation of Mindfulness-Based iPhone Apps”, JMIR mHealth and uHealth, vol. 3, no. 3, p. e82, 2015. Available: 10.2196/mhealth.4328.

[19] J. Flett, H. Hayne, B. Riordan, L. Thompson and T. Conner, “Mobile Mindfulness Meditation: a Randomised Controlled Trial of the Effect of Two Popular Apps on Mental Health”, Mindfulness, vol. 10, no. 5, pp. 863-876, 2018. Available: 10.1007/s12671-018-1050-9 [Accessed 22 August 2022].

[20] B. Grepon, N. Baran, K. Gumonan, A. Martinez and M. Lacsa, “Designing and Implementing e-School Systems: An Information Systems Approach to School Management of a Community College in Northern Mindanao, Philippines”, International Journal of Computing Sciences Research, vol. 6, pp. 792-808, 2022. Available: 10.25147/ijcsr.2017.001.1.74.

[21] A. Rasnacis and S. Berzisa, “Method for Adaptation and Implementation of Agile Project Management Methodology”, Procedia Computer Science, vol. 104, pp. 43-50, 2017. Available: 10.1016/j.procs.2017.01.055.

[22] K. Pathak, “7 meditation apps that are cheaper (and better) than headspace and calm,” Lifehacker, 13-Aug-2021. [Online]. Available: https://lifehacker.com/7-meditation-apps-that-are-cheaper-and-better-than-he-1847455111. [Accessed: 14-Apr-2023].

[23] M. E. Sara Lindberg, “How your environment affects your mental health,” Verywell Mind, 23-Mar-2023. [Online]. Available: https://www.verywellmind.com/how-your-environment-affects-your-mental-health-5093687. [Accessed: 14-Apr-2023]. [24]I. Balagam, “Is headspace right for you? here's what our experts have to say,” Healthline, 29-Jan-2022. [Online]. Available: https://www.healthline.com/health/mental-health/headspace-review. [Accessed: 14-Apr-2023].

[25] “Firebase - Introduction,” GeeksforGeeks, 15-Jul-2021. [Online]. Available: https://www.geeksforgeeks.org/firebase-introduction/. [Accessed: 19-Apr-2023].

[26] M. C. Pascoe, S. E. Hetrick, and A. G. Parker, “The impact of stress on students in Secondary School and Higher Education,” International Journal of Adolescence and Youth, vol. 25, no. 1, pp. 104–112, 2019. doi:10.1080/02673843.2019.1596823

[27] Sophie Bethune, “Demand for mental health treatment continues to increase, say psychologists,” American Psychological Association, https://www.apa.org/news/press/releases/2021/10/mental-health-treatment-demand (accessed Aug. 16, 2023).

[28] E. Scott, "What are the main causes of stress?," The Main Causes of Stress, https://www.verywellmind.com/what-are-the-main-causes-of-stress-3145063 (accessed Aug. 18, 2023).

[29] E. Scott, "How chronic stress impacts your health," How Chronic Stress Impacts Your Health, https://www.verywellmind.com/chronic-stress-3145104 (accessed Aug. 18, 2023).

[30] L. Juul, K. J. Pallesen, M. Bjerggaard, C. Nielsen, and L. O. Fjorback, "A pilot randomised trial comparing a mindfulness-based stress reduction course, a locally-developed stress reduction intervention and a waiting list control group in a real-life municipal health care setting," *BMC Public Health*, vol. 20, no. 1, 2020. doi:10.1186/s12889-020-08470-6

[31] B. Marie, "A gentle introduction to GPT models," Medium, https://towardsdatascience.com/a-gentle-introduction-to-gpt-models-e02b093a495b (accessed Sep. 6, 2023).

[32] OpenAI, "OpenAI platform," OpenAI Models, https://platform.openai.com/docs/models (accessed Sep. 6, 2023).

[33] A. Mandour, "GPT-3.5 model architecture," OpenGenus IQ: Computing Expertise & Legacy, https://iq.opengenus.org/gpt-3-5-model/ (accessed Sep. 6, 2023).

[34] Q. Kong, T. Siauw, and A. Bayen, "Pickle files — python numerical methods - university of california ...," Contents Table of Contents Python Programming And Numerical Methods: A Guide For Engineers And Scientists¶, https://pythonnumericalmethods.berkeley.edu/notebooks/chapter11.03-Pickle-Files.html (accessed Sep. 6, 2023).

[35] "What is the K-nearest neighbors algorithm?," IBM, https://www.ibm.com/topics/knn (accessed Sep. 12, 2023).

[36] G. Seif, "A beginner's guide to xgboost," Medium, https://towardsdatascience.com/a-beginners-guide-to-xgboost-87f5d4c30ed7 (accessed Sep. 12, 2023).

[37] T. Yiu, “Understanding random forest,” Medium, https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (accessed Sep. 12, 2023).

[38] “What is a decision tree,” IBM, https://www.ibm.com/topics/decision-trees (accessed Sep. 12, 2023).

[39] V. Rattan, R. Mittal, J. Singh, and V. Malik, “Analyzing the application of smote on machine learning classifiers,” *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, 2021. doi:10.1109/esci50559.2021.9396962

[40] A. Band, “How to find the optimal value of K in Knn?,” Medium, https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb (accessed Sep. 12, 2023).

[41] “XGBoost parameters,” XGBoost Parameters - xgboost 2.1.0-dev documentation, https://xgboost.readthedocs.io/en/latest/parameter.html (accessed Sep. 13, 2023).

[42] T. Keldenich, “Decision tree how to use it and its hyperparameters,” Inside Machine Learning, https://inside-machinelearning.com/en/decision-tree-and-hyperparameters/ (accessed Sep. 13, 2023).

[43] “Sklearn.ensemble.randomforestclassifier,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed Sep. 13, 2023).

[44] “Sklearn.neighbors.kneighborsclassifier,” scikit, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (accessed Sep. 13, 2023).

APPENDIX 1

DASS-21 Questionnaire Sample

1. I found it hard to wind down.
A. Did not apply to me at all.
B. Applied to me to some degree or some of the time.
C. Applied to me to a considerable degree or a good part of the time.
D. Applied to me very much or most of the time.

2. I tended to over-react to situation.
A. Did not apply to me at all.
B. Applied to me to some degree or some of the time.
C. Applied to me to a considerable degree or a good part of the time.
D. Applied to me very much or most of the time.

3. I felt that I was using a lot of nervous energy.
A. Did not apply to me at all.
B. Applied to me to some degree or some of the time.
C. Applied to me to a considerable degree or a good part of the time.
D. Applied to me very much or most of the time.

4. I found myself getting agitated.
A. Did not apply to me at all.
B. Applied to me to some degree or some of the time.
C. Applied to me to a considerable degree or a good part of the time.
D. Applied to me very much or most of the time.

5. I found it difficult to relax.
A. Did not apply to me at all.
B. Applied to me to some degree or some of the time.
C. Applied to me to a considerable degree or a good part of the time.
D. Applied to me very much or most of the time.

6. I was intolerant of anything that kept me from getting on with what I was doing.
A. Did not apply to me at all.
B. Applied to me to some degree or some of the time.
C. Applied to me to a considerable degree or a good part of the time.
D. Applied to me very much or most of the time.

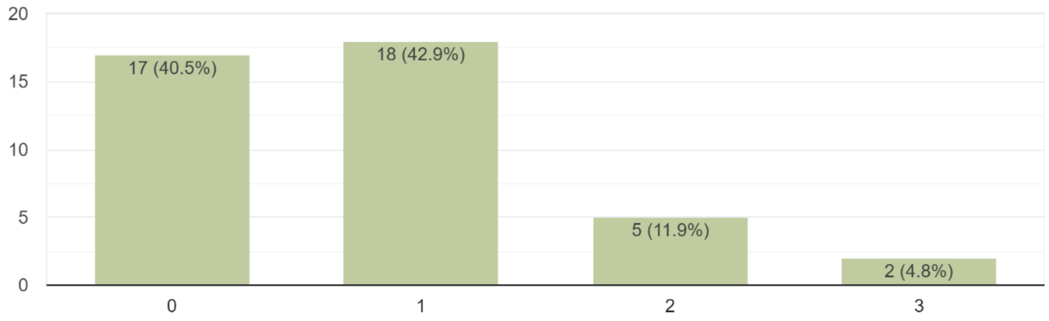
7. I felt that I was rather touchy.
A. Did not apply to me at all.
B. Applied to me to some degree or some of the time.
C. Applied to me to a considerable degree or a good part of the time.
D. Applied to me very much or most of the time.

APPENDIX 2

DASS-21 Questionnaire Responses

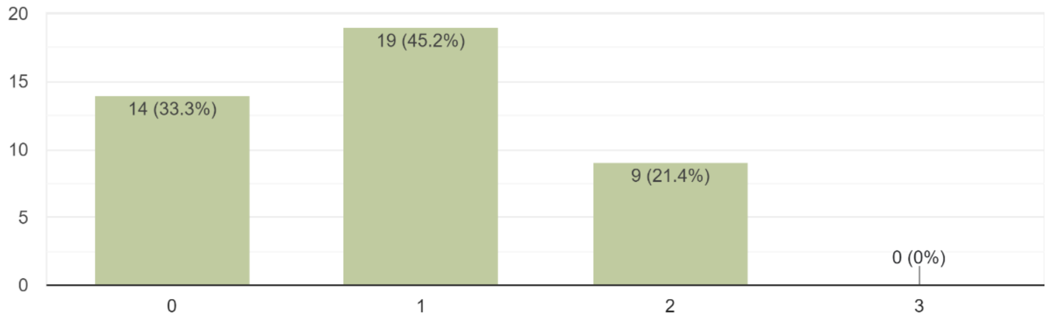
1. I found it hard to wind down.

42 responses



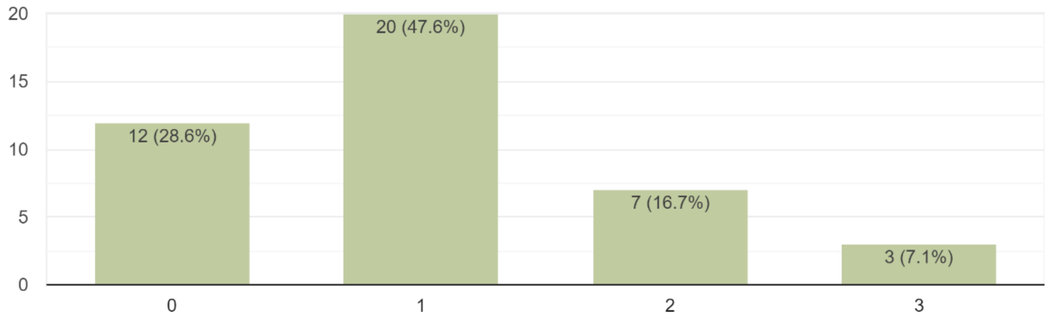
6. I tended to over-react to situations.

42 responses



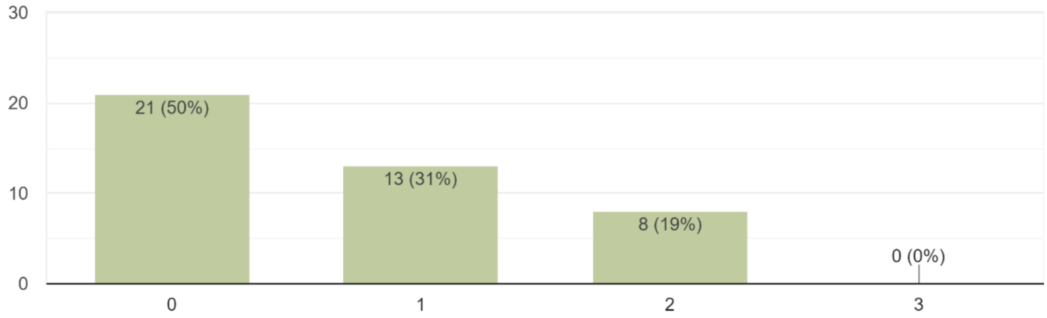
8. I felt that I was using a lot of nervous energy.

42 responses



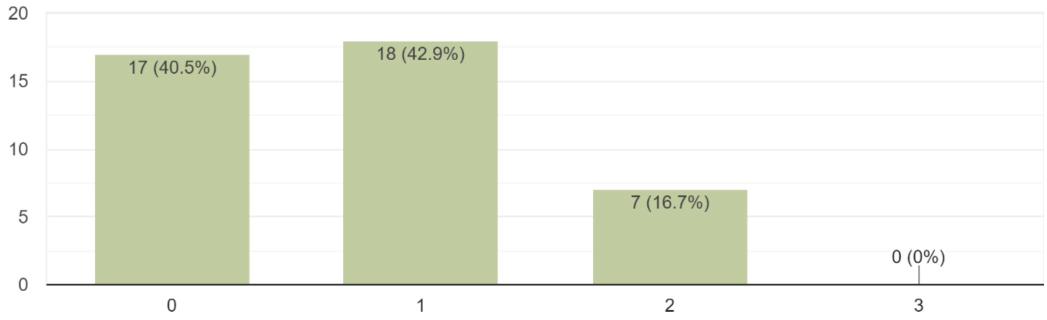
11. I found myself getting agitated.

42 responses



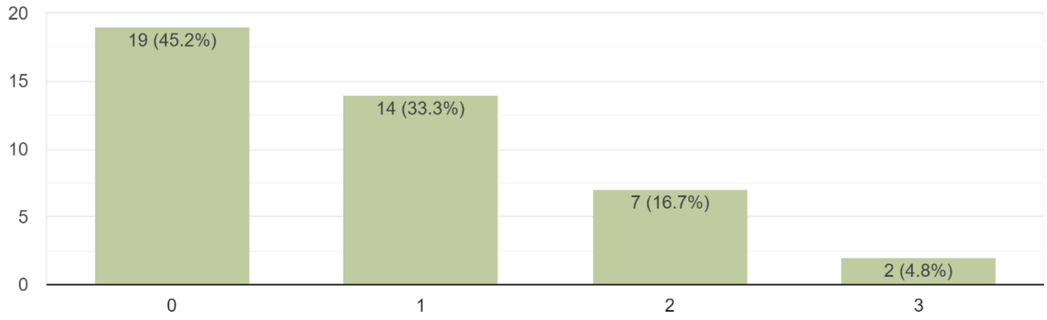
12. I found it difficult to relax.

42 responses



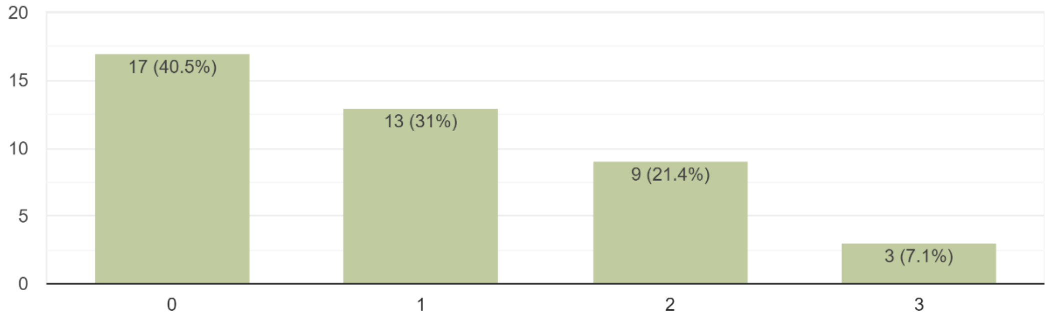
14. I was intolerant of anything that kept me from getting on with what I was doing.

42 responses



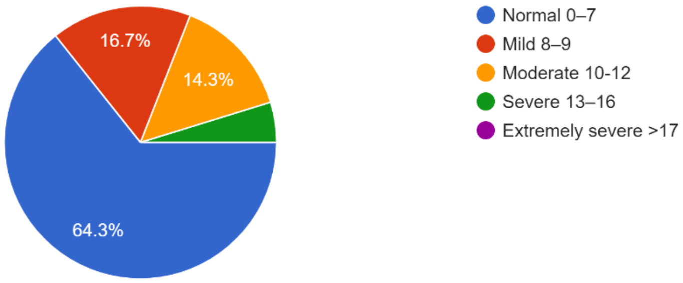
18. I felt that I was rather touchy.

42 responses



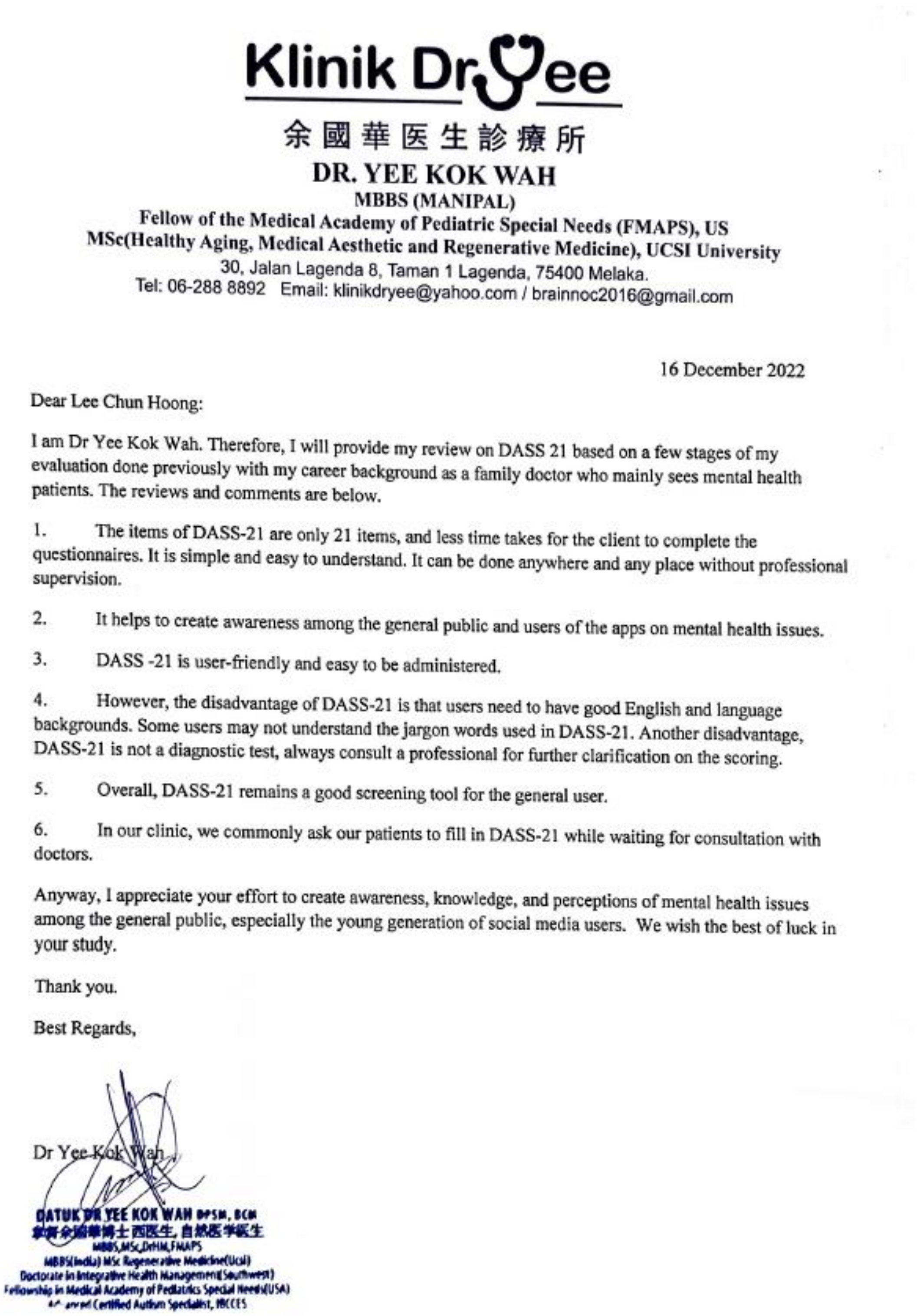
Total Scores of Stress Scale

42 responses



APPENDIX 3

Supporting letter of DASS-21



Klinik Dr Yee

余國華医生診療所

DR. YEE KOK WAH

MBBS (MANIPAL)

Fellow of the Medical Academy of Pediatric Special Needs (FMAPS), US

MSc(Healthy Aging, Medical Aesthetic and Regenerative Medicine), UCSI University

30, Jalan Lagenda 8, Taman 1 Lagenda, 75400 Melaka.

Tel: 06-288 8892 Email: klinikdryee@yahoo.com / brainnoc2016@gmail.com

16 December 2022

Dear Lee Chun Hoong:

I am Dr Yee Kok Wah. Therefore, I will provide my review on DASS 21 based on a few stages of my evaluation done previously with my career background as a family doctor who mainly sees mental health patients. The reviews and comments are below.

1. The items of DASS-21 are only 21 items, and less time takes for the client to complete the questionnaires. It is simple and easy to understand. It can be done anywhere and any place without professional supervision.

2. It helps to create awareness among the general public and users of the apps on mental health issues.

3. DASS -21 is user-friendly and easy to be administered.

4. However, the disadvantage of DASS-21 is that users need to have good English and language backgrounds. Some users may not understand the jargon words used in DASS-21. Another disadvantage, DASS-21 is not a diagnostic test, always consult a professional for further clarification on the scoring.

5. Overall, DASS-21 remains a good screening tool for the general user.

6. In our clinic, we commonly ask our patients to fill in DASS-21 while waiting for consultation with doctors.

Anyway, I appreciate your effort to create awareness, knowledge, and perceptions of mental health issues among the general public, especially the young generation of social media users. We wish the best of luck in your study.

Thank you.

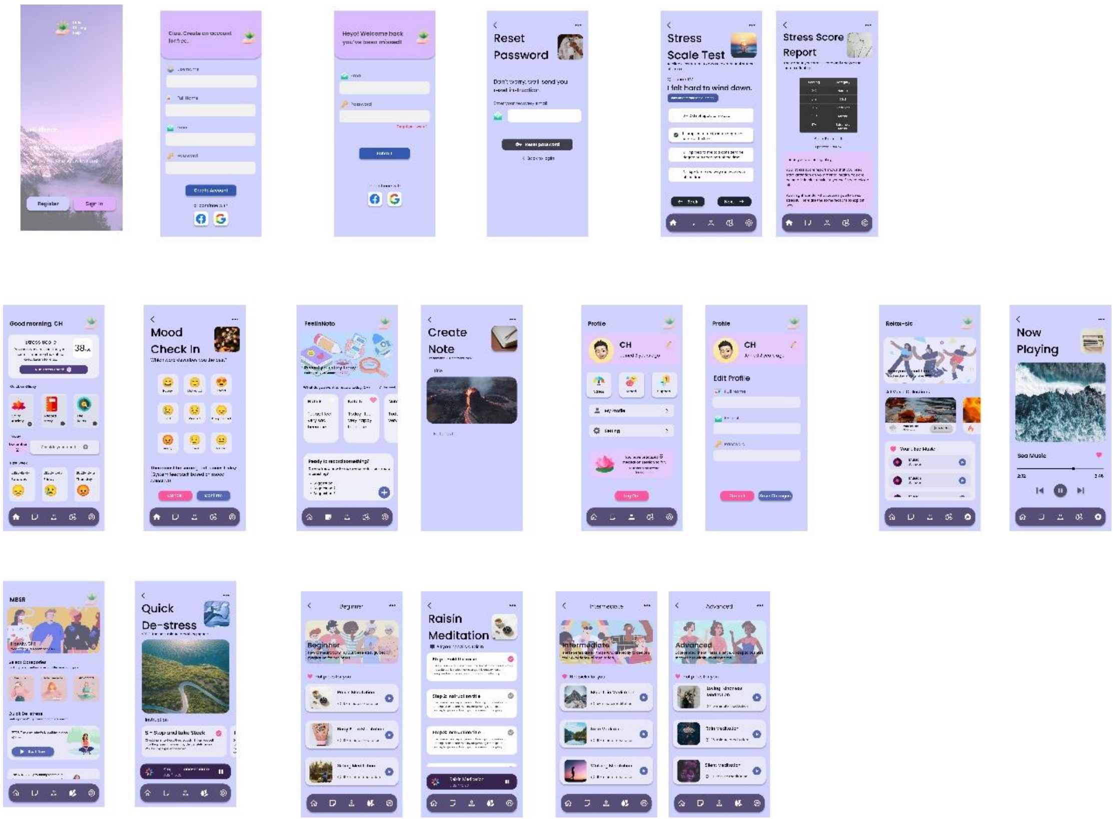
Best Regards,

Dr Yee Kok Wah

DATUK DR YEE KOK WAH DPSM, BCM
拿督余國華博士 西医生, 自然医学医生
MBBS,MSc,DrHM,FMAPS
MBBS(India) MSc Regenerative Medicine(Ucsi)
Doctorate in Integrative Health Management(Southwest)
Fellowship in Medical Academy of Pediatrics Special Needs(USA)
Advanced Certified Autism Specialist, IBCCES

APPENDIX 4

Application Interface Design



APPENDIX 5

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 2**
Student Name & ID: LEE CHUN HOONG 19ACB05878	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Stress Mental Health Symptom Assessment Mobile Application for Young Adults	

1. WORK DONE

- Completed the development of MBSR Meditation feature.
- Inserted a total of 9 meditation exercises ranging from Beginner, Intermediate, and Advanced.

2. WORK TO BE DONE

- Develop both text and audio instruction for each meditation exercise so users can learn conveniently.
- Set up Expo Audio library to load audio file into the feature.

3. PROBLEMS ENCOUNTERED

In specific circumstances, the text instructions for meditation may be overly lengthy and complex, require simplifications while retaining the original intended meaning.

4. SELF EVALUATION OF THE PROGRESS

The meditation exercise can be selected and users can choose to follow the text instruction or audio instruction based on their preference.

V. mgh

Supervisor's signature

ChunHoong

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 4**
Student Name & ID: LEE CHUN HOONG 19ACB05878	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Stress Mental Health Symptom Assessment Mobile Application for Young Adults	

1. WORK DONE

- Several MBSR calming music were selected into the music playlist.
- The music playlist is then implemented into the MBSR Relaxing Music feature, allowing users to choose the music to be played.
- The music player was inspired by Spotify application to design the similar application interface and functions.

2. WORK TO BE DONE

Learning most of the functions under the Expo Audio and Expo Video Library to develop the music player interface.

3. PROBLEMS ENCOUNTERED

The YouTube Player API was initially utilized to import music videos from YouTube. However, due to compatibility issues between the API version and React Native, the process of successfully importing video files from YouTube failed. Consequently, the errors encountered prompted the adoption of the Expo Video library, enabling the direct insertion of video files into the application.

4. SELF EVALUATION OF THE PROGRESS

Expo Video library offers many functionalities which are simple to use and resources efficient. The music player was successfully developed by utilizing both Expo Audio and Expo Video library.

Chun Hoong

Supervisor's signature   Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 6**
Student Name & ID: LEE CHUN HOONG 19ACB05878	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Stress Mental Health Symptom Assessment Mobile Application for Young Adults	

1. WORK DONE

- Program the interface of Note Taking feature.
- Able to store newly created note and update it on the database.
- Retrieve existing created notes from the database and display it to the user for viewing.

2. WORK TO BE DONE

- Program the profile page of the application.
- Develop profile editing function.
- Start learning about the GPT model used to develop the chatbot application.

3. PROBLEMS ENCOUNTERED

The note-taking feature must be adaptable to various smartphone screen dimensions within the application. Hence, the Keyboard Height parameter requires frequent adjustments to attain the optimal height, ensuring that users can comfortably view the complete note description without excessive scrolling.

4. SELF EVALUATION OF THE PROGRESS

React Native is easy to implement when developing the note taking feature as it can prevent errors during saving a new note to the Firebase database by using the “await” keyword. This keyword is typically used in asynchronous programming to wait for a promise to resolve. It means that the code execution will pause at this point until the promise is resolved.

ChunHoong

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 8**
Student Name & ID: LEE CHUN HOONG 19ACB05878	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Stress Mental Health Symptom Assessment Mobile Application for Young Adults	

1. WORK DONE

- Program the mood tracker interface.
- Saves the updated mood list into the database.
- Auto refresh the dashboard to display the latest mood list.

2. WORK TO BE DONE

- Gather MBSR resources which includes the instruction lists and audio instruction.
- Program MBSR Quick Destress function.

3. PROBLEMS ENCOUNTERED

By using “useEffect” to make sure the dashboard can refresh itself once the users update the mood of the day, it might cause the page to refresh infinitely which increase the frequency of documents read in the database and cause the workload to be increased.

4. SELF EVALUATION OF THE PROGRESS

The dashboard of the application is almost completed to display the current stress score and moods experienced for the past seven days. Users can also access to MBSR programs through dashboard for quick solutions.

V.mgh ______________________ Supervisor’s signature

ChunHoong ______________________ Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T2	**Study week no.: 10**
Student Name & ID: LEE CHUN HOONG 19ACB05878	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Stress Mental Health Symptom Assessment Mobile Application for Young Adults	

1. WORK DONE

- Successfully developed the mini-AI chatbot functions.
- Load the default welcoming messages properly when users interact with the chatbot.
- Integrate GPT model into the application to act as a conversational agent.

2. WORK TO BE DONE

- Develop the prediction model using KNN classifier.
- Writing the FYP2 report.

3. PROBLEMS ENCOUNTERED

The GPT model used has certain number of limits to send and receive responses so the API secret key must be replaced constantly. Moreover, the GPT model sometimes will encounter the "Axios network error" bug where it cannot connect to the API services.

4. SELF EVALUATION OF THE PROGRESS

The work planned in FYP2 is almost complete, including accomplishing few objectives stated in my proposal report. However, more functions and activities to relieve stress need to be developed to increase the effectiveness of the application.

Supervisor's signature

ChunHoong

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3T3	**Study week no.: 12**
Student Name & ID: LEE CHUN HOONG 19ACB05878	
Supervisor: Dr Mogana a/p Vadiveloo	
Project Title: Stress Mental Health Symptom Assessment Mobile Application for Young Adults	

1. WORK DONE

- Complete the development of KNN prediction tool.
- Finalize every function of FYP2 and the MBSR.
- Report writing on progress.

2. WORK TO BE DONE

- Prepare for presentation and demonstration.

3. PROBLEMS ENCOUNTERED

The current code is mixed with the styling code and need to be refactored in another code section to make the code more readable.

4. SELF EVALUATION OF THE PROGRESS

I learned about the development of frontend and backend services especially make use of the functions in Google Firebase Firestore database, integration of GPT model API, and developing own classification model. It improves my programming problem solving technique during the project development.

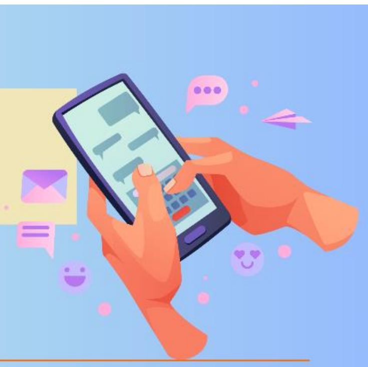
V. | ChunHoong
Supervisor's signature | Student's signature

APPENDIX 6

POSTER

STRESS MENTAL HEALTH SYMPTOM ASSESSMENT MOBILE APPLICATION FOR YOUNG ADULTS

This proposed application will allow users to conduct a stress assessment session in the application inspired by Mindfulness-Based Stress Reduction Program. A series of practical activities are recommended to the users to relieve the stress. Moreover, users can provide more inputs based on their feelings throughout the day.



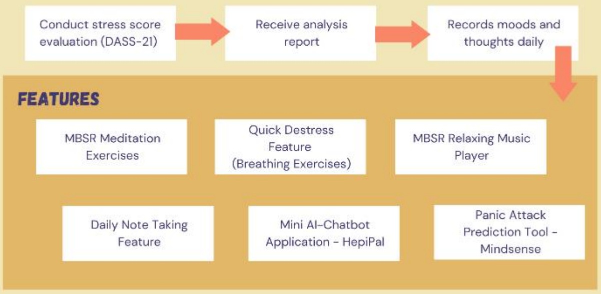
INTRODUCTION

- Help young adults to identify and overcome stress in everyday life.
- Develop mHealth application with different functionalities to reduce stress.
- Implement a Mindfulness-Based Stress Reduction Program to develop the features to help mitigate stress

OBJECTIVE

- Provide stress assessment sessions.
- Integrate Mindfulness-Based Stress Reduction.
- Encourage users to express thoughts with an integrated AI chatbot.
- Develop a panic attack prediction tool with machine learning algorithm.

METHODS



Conduct stress score evaluation (DASS-21)

Receive analysis report

Records moods and thoughts daily

FEATURES

MBSR Meditation Exercises

Quick Destress Feature (Breathing Exercises)

MBSR Relaxing Music Player

Daily Note Taking Feature

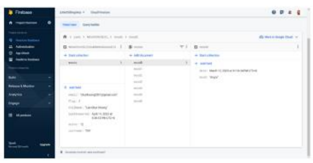
Mini AI-Chatbot Application – HepiPal

Panic Attack Prediction Tool – Mindsense

ANALYSIS

The proposed application (Little Chilling Help) is developed under the React Native framework and agile methodology.

- Apply the MBSR course material as one of the functionalities by encouraging the users to practice a good state of awareness to reduce stress.
- Include an AI chatbot using the GPT model API from OpenAI that allows users to express their feelings whenever and wherever they feel like it or even consult advice from the chatbot.
- Implement a K-Nearest Neighbour (KNN) classifier to build the panic attack prediction model as KNN achieves the best testing accuracy of 70.31% among other models



Google Firebase Firestore is established to store important user and program data

HepiPal

Mini AI Chatbot based on the GPT model

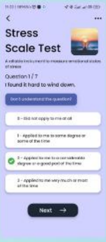
MindSense For Students

Panic attack prediction tool developed by KNN classifier

CONCLUSION

The proposed application is designed to offer users a comprehensive set of features for evaluating, monitoring, and managing their stress levels over time. It encourages users to engage with their emotions and thoughts, which are key components of mindfulness in MBSR practices.

RESULT



Stress Scale Test

Question 1 / 7

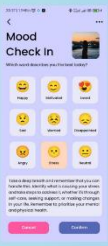
I found it hard to wind down.

0 - Did not apply to me at all

1 - Applied to me to some degree or some of the time

2 - Applied to me to a considerable degree or a good part of the time

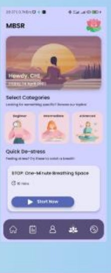
3 - Applied to me very much or most of the time



Mood Check In

Which word describes you the best today?

Take a deep breath and remember that you can handle this. Identify what is causing your stress and take steps to address it, whether it's through self-care, seeking support, or making changes in your life. Remember to prioritize your mental and physical health.



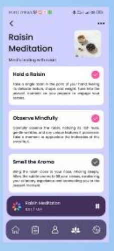
MBSR

Howdy, Chi

Select Categories

Quick De-stress

STOP: One-Minute Breathing Space

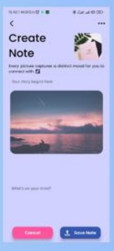


Raisin Meditation

Hold a Raisin

Observe Mindfully

Smell the Aroma



Create Note

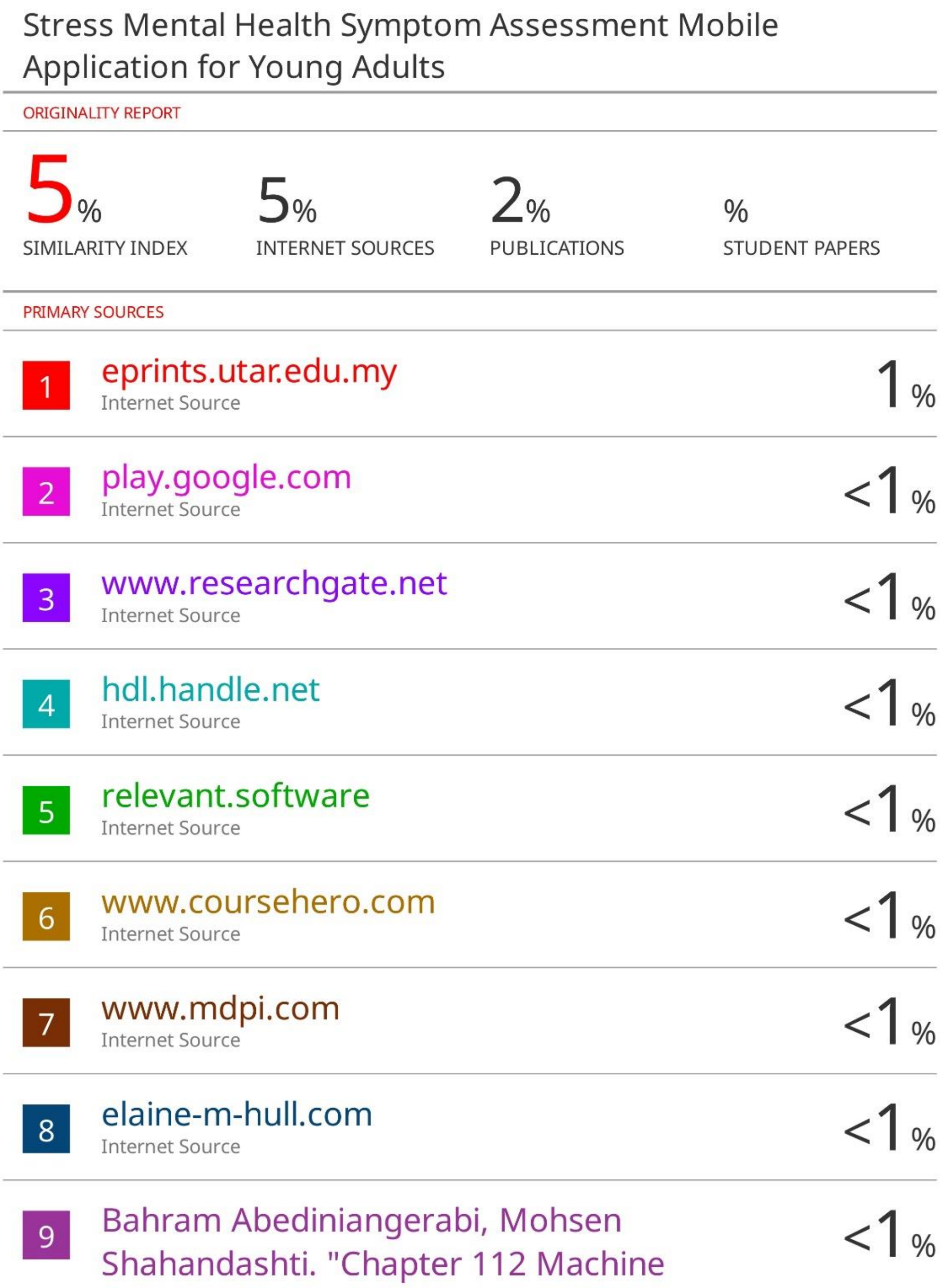
Every picture captures a distinct mood for you to connect with.

What's on your mind?

Lee Chun Hoong
1905878

Bachelor of Computer Science (Hons)
Final Year Project

PLAGIARISM CHECK RESULT



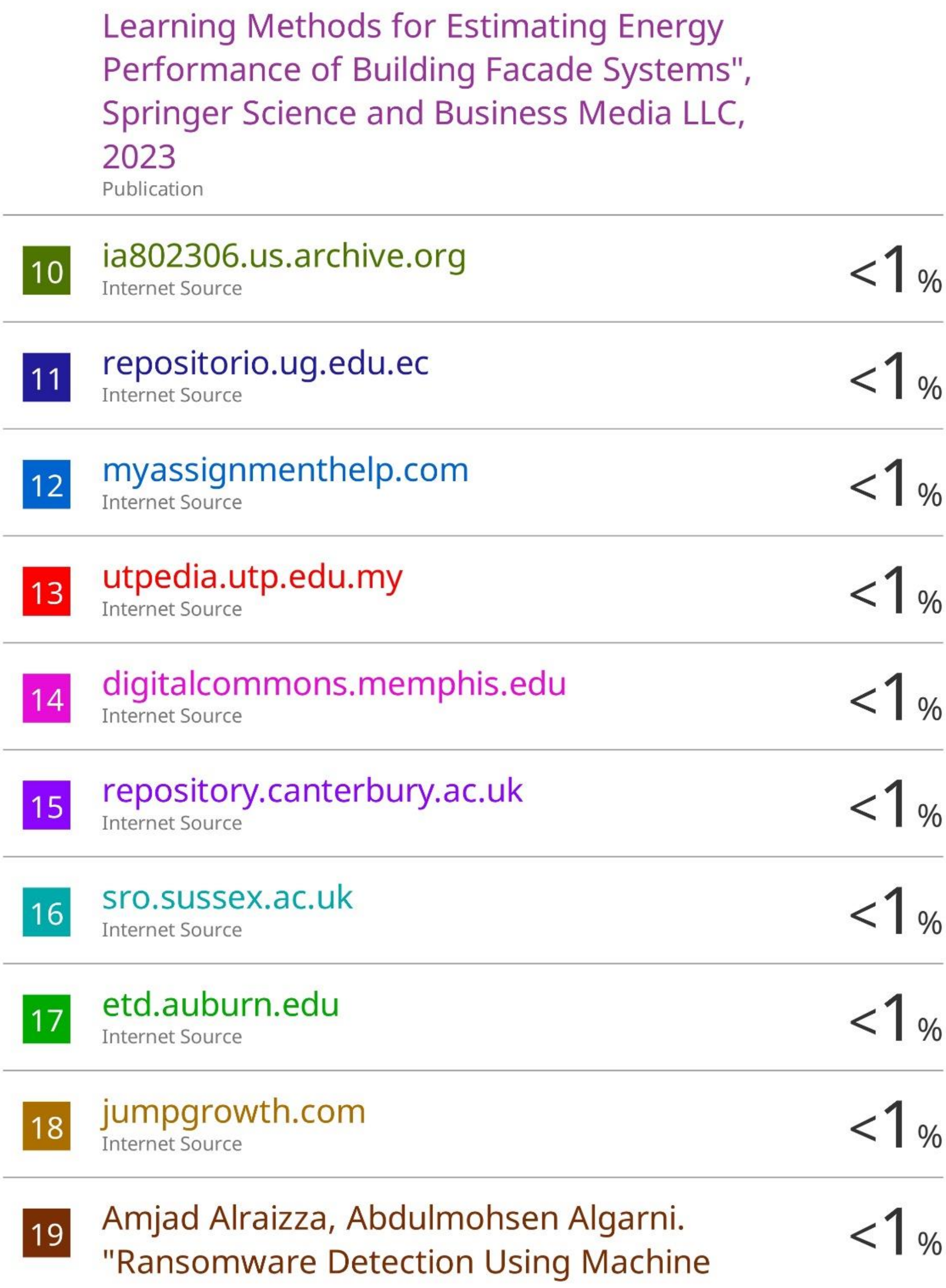
Stress Mental Health Symptom Assessment Mobile Application for Young Adults

ORIGINALITY REPORT

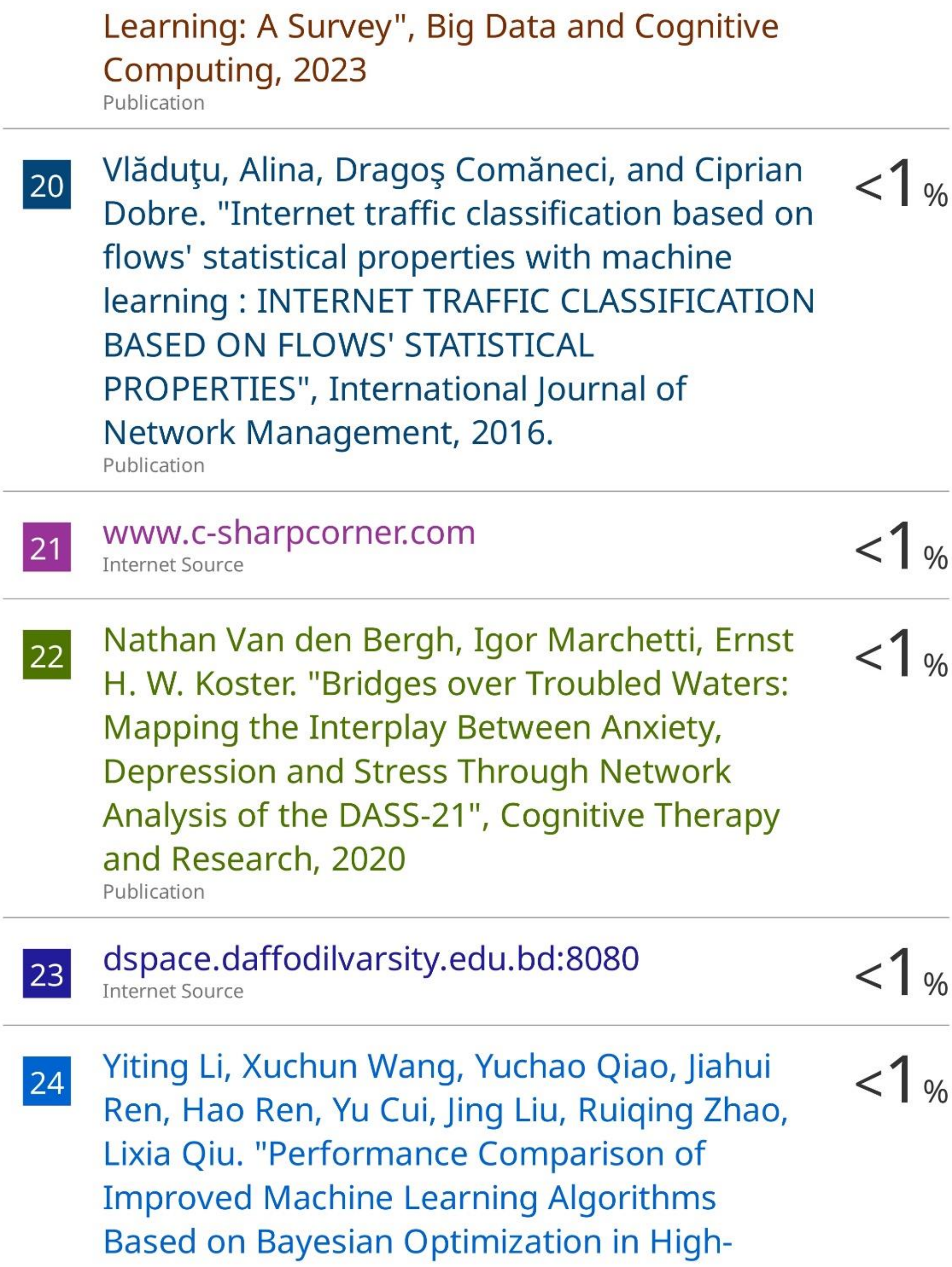
5%	5%	2%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

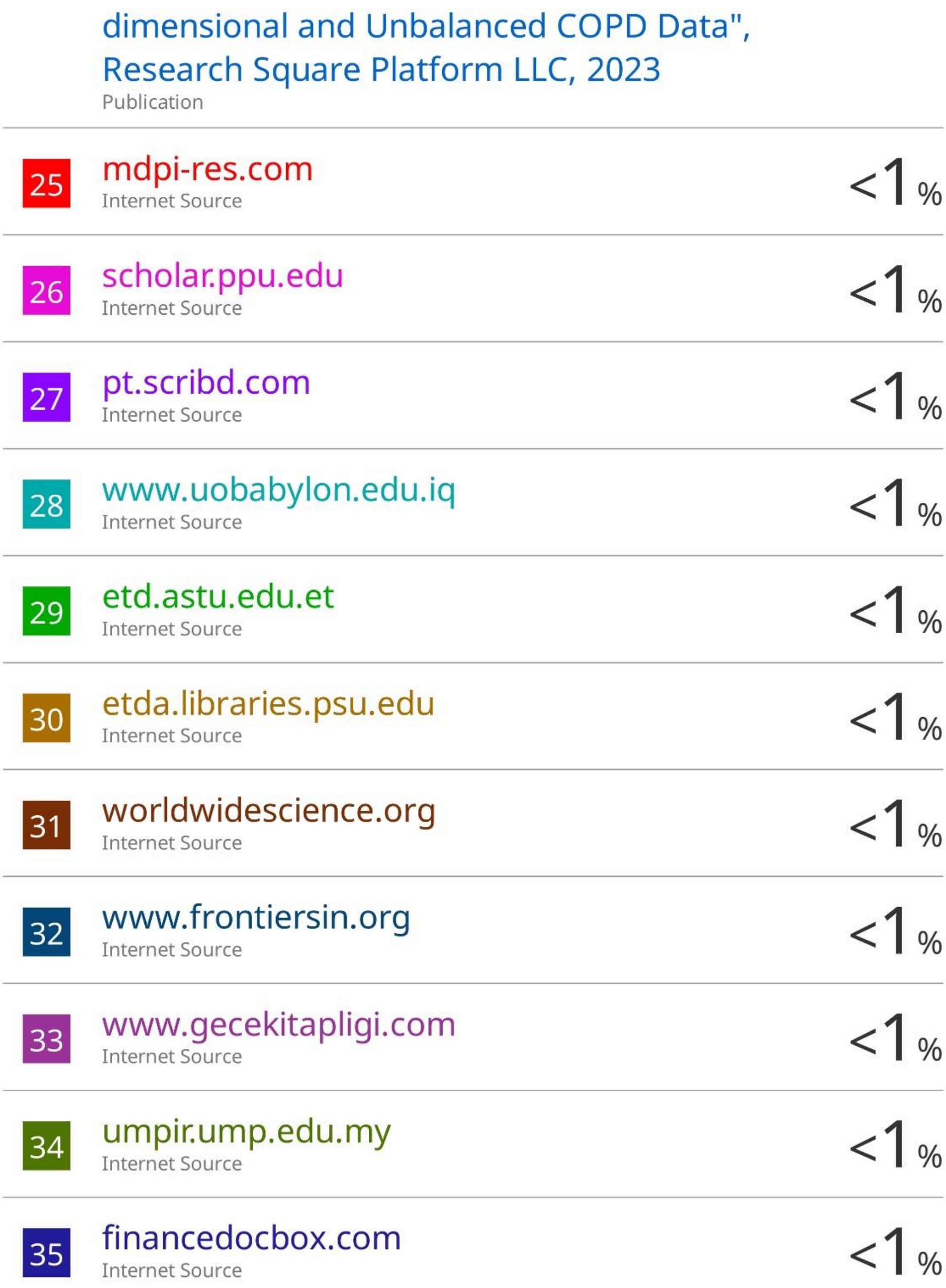
1	eprints.utar.edu.my Internet Source	1 %
2	play.google.com Internet Source	<1 %
3	www.researchgate.net Internet Source	<1 %
4	hdl.handle.net Internet Source	<1 %
5	relevant.software Internet Source	<1 %
6	www.coursehero.com Internet Source	<1 %
7	www.mdpi.com Internet Source	<1 %
8	elaine-m-hull.com Internet Source	<1 %
9	Bahram Abediniangerabi, Mohsen Shahandashti. "Chapter 112 Machine	<1 %



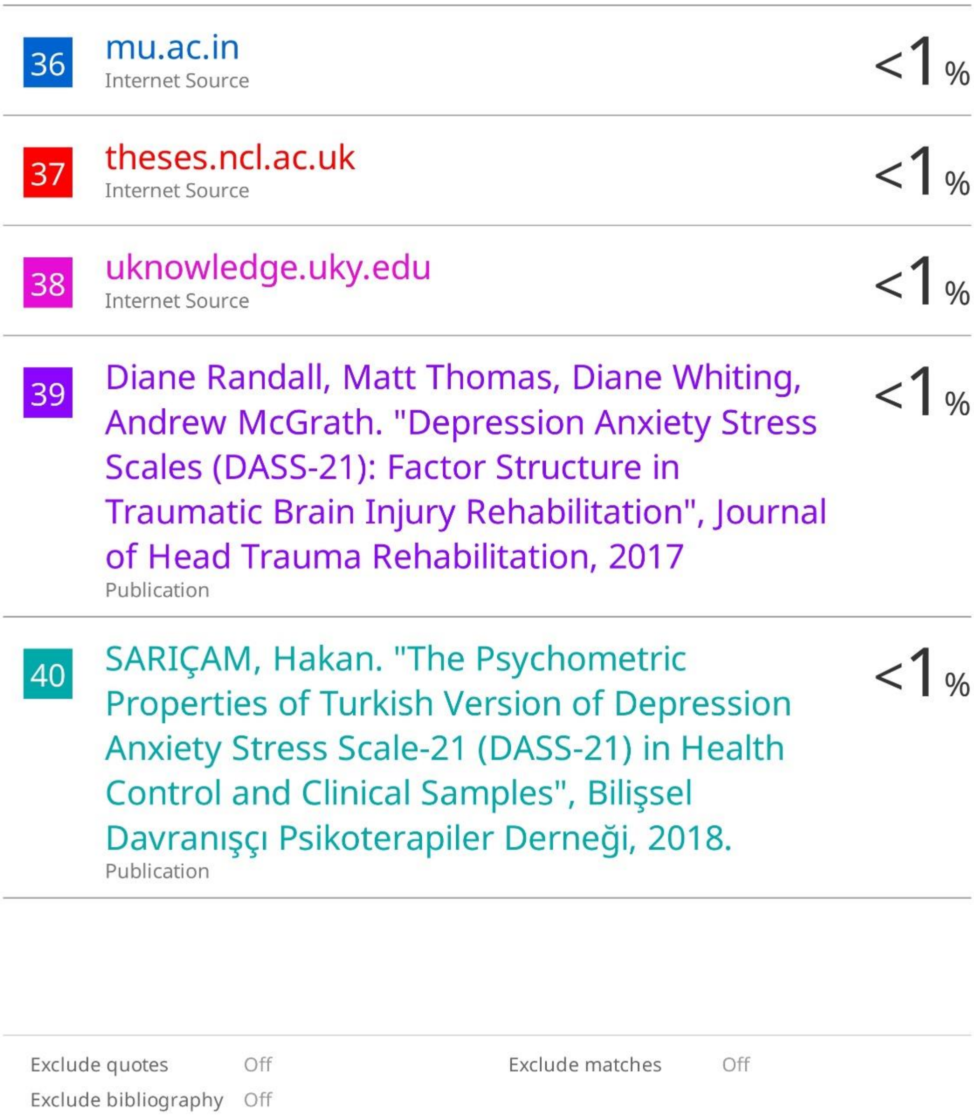
	Learning Methods for Estimating Energy Performance of Building Facade Systems", Springer Science and Business Media LLC, 2023 Publication	
10	ia802306.us.archive.org Internet Source	<1 %
11	repositorio.ug.edu.ec Internet Source	<1 %
12	myassignmenthelp.com Internet Source	<1 %
13	utpedia.utp.edu.my Internet Source	<1 %
14	digitalcommons.memphis.edu Internet Source	<1 %
15	repository.canterbury.ac.uk Internet Source	<1 %
16	sro.sussex.ac.uk Internet Source	<1 %
17	etd.auburn.edu Internet Source	<1 %
18	jumpgrowth.com Internet Source	<1 %
19	Amjad Alraizza, Abdulmohsen Algarni. "Ransomware Detection Using Machine	<1 %



	Learning: A Survey", Big Data and Cognitive Computing, 2023 Publication	
20	Vlăduţu, Alina, Dragoş Comăneci, and Ciprian Dobre. "Internet traffic classification based on flows' statistical properties with machine learning : INTERNET TRAFFIC CLASSIFICATION BASED ON FLOWS' STATISTICAL PROPERTIES", International Journal of Network Management, 2016. Publication	<1 %
21	www.c-sharpcorner.com Internet Source	<1 %
22	Nathan Van den Bergh, Igor Marchetti, Ernst H. W. Koster. "Bridges over Troubled Waters: Mapping the Interplay Between Anxiety, Depression and Stress Through Network Analysis of the DASS-21", Cognitive Therapy and Research, 2020 Publication	<1 %
23	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
24	Yiting Li, Xuchun Wang, Yuchao Qiao, Jiahui Ren, Hao Ren, Yu Cui, Jing Liu, Ruiqing Zhao, Lixia Qiu. "Performance Comparison of Improved Machine Learning Algorithms Based on Bayesian Optimization in High-	<1 %



	dimensional and Unbalanced COPD Data", Research Square Platform LLC, 2023 Publication	
25	mdpi-res.com Internet Source	<1 %
26	scholar.ppu.edu Internet Source	<1 %
27	pt.scribd.com Internet Source	<1 %
28	www.uobabylon.edu.iq Internet Source	<1 %
29	etd.astu.edu.et Internet Source	<1 %
30	etda.libraries.psu.edu Internet Source	<1 %
31	worldwidescience.org Internet Source	<1 %
32	www.frontiersin.org Internet Source	<1 %
33	www.gecekitapligi.com Internet Source	<1 %
34	umpir.ump.edu.my Internet Source	<1 %
35	financedocbox.com Internet Source	<1 %



36	mu.ac.in Internet Source	<1 %
37	theses.ncl.ac.uk Internet Source	<1 %
38	uknowledge.uky.edu Internet Source	<1 %
39	Diane Randall, Matt Thomas, Diane Whiting, Andrew McGrath. "Depression Anxiety Stress Scales (DASS-21): Factor Structure in Traumatic Brain Injury Rehabilitation", Journal of Head Trauma Rehabilitation, 2017 Publication	<1 %
40	SARIÇAM, Hakan. "The Psychometric Properties of Turkish Version of Depression Anxiety Stress Scale-21 (DASS-21) in Health Control and Clinical Samples", Bilişsel Davranışçı Psikoterapiler Derneği, 2018. Publication	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	LEE CHUN HOONG
ID Number(s)	19ACB05878
Programme / Course	Bachelor in Computer Science (Honours)
Title of Final Year Project	Stress Mental Health Symptom Assessment Mobile Application for Young Adults

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 5 **%** **Similarity by source** Internet Sources: 5 % Publications: 2 % Student Papers: 0 %	No comment.
Number of individual sources listed of more than 3% similarity: -	No comment.

Parameters of originality required, and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Dr. Mogana a/p Vadiveloo	Name:
Date: 15/09/2023	Date:

FYP 2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB05878
Student Name	LEE CHUN HOONG
Supervisor Name	Dr Mogana a/p Vadiveloo

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
X	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

ChunHoong

(Signature of Student)
Date: 15/09/2023