

Design Project Report

M11 Design Project (2023-2A)

MindfulStudent: A Student Mental Health App
Group 5

Authors:

Umar Kashif	s277235	<i>m.u.kashif@student.utwente.nl</i>
Mike Almeloo	s2809230	<i>m.almeloo@student.utwente.nl</i>
Aarav Ashish Mehrishi	s2477521	<i>a.a.mehrishi@student.utwente.nl</i>
Batuhan Usta	s2807025	<i>b.usta@student.utwente.nl</i>
Kristian Guenov	s2711001	<i>k.guenov@student.utwente.nl</i>
Veselin Shterev	s2503921	<i>v.shterev@student.utwente.nl</i>

Supervisor:

dr.ir. M.Zangiabady (Mahboobeh) *m.zangiabady@utwente.nl*

**UNIVERSITY
OF TWENTE.**

“

Mental health ... is not a destination, but a process. It's about how you drive, not where you're going.

“

- Noam Shpancer, PHD

Table of Contents

Chapter 1: Background	6
1.1 The Identified Problem and The Solution	6
1.2 Purpose, Scope, and Applicability	7
1.2.1 Purpose	7
1.2.2 Scope	7
1.2.3 Applicability	8
1.3 Achievements	8
Chapter 2: Project Introduction	10
2.1 The Client	10
2.2 Direct Stakeholders	10
2.2.1 Students	10
2.2.2 Mental Health Experts	10
2.2.3 Emergency Organizations	10
2.3 Indirect Stakeholders	11
2.2.1 Universities	11
2.2.2 Organisations	11
2.4 Requirements Specification	11
2.4.1 Account System	11
2.4.2 Sleep Improvement	12
2.4.3 Peer Communication	12
2.4.4 Mental Health Expert Communication	12
2.4.5 Breathing Exercises	13
2.4.6 Digital Journal	13
2.4.7 Guided Sessions	13
2.4.8 Gamification	13
2.4.9 Emergency Information	13
2.4.10 Book Tracking	14
2.4.11 Progress Tracking	14
2.4.12 Privacy and Security	14
2.5 Use Case Diagram	14
2.6 Software and Hardware Requirements	15
2.7 Teamwork and Planning	16
Chapter 3: User Interface Design	19
3.1 Initial Design	19
3.2 Final Design	19

3.3 Design Philosophy	19
3.4 Design improvements	20
Chapter 4: Ethical and Security Considerations	21
4.1 Ethics	21
4.2 Security	21
4.3 Privacy	22
Chapter 5: Implementation	24
5.1 Technology Stack	24
5.2 Implementation Details	25
5.2.1 Authentication	26
5.2.2 Database	26
5.2.3 Supabase Edge Functions	28
5.2.4 Sleep Tracking	29
5.2.5 Push Notifications	32
5.2.6 Flutter Data Integration	34
Chapter 6: Testing	37
6.1 User Testing	37
6.2 System Testing	37
6.3 Modifications and Improvements	40
Chapter 7: Conclusions	41
7.1 Key Achievements	41
7.2 Limitations of the System	43
7.3 Future Scope of the Project	44
References	46
Glossary	47
Appendix	49
A - Initial Design	49
B - Final Design	51
C - Discarded UI Pages	53
D - User Survey	53
E - Usability Study	67

Chapter 1: Background

In this first chapter, we will provide an introduction to the topic of mental health among university students. Its goal is to provide a background into what makes this particular project unique with its elements that can provide significant improvements to the lives of students worldwide. It outlines the problem, the solution, the purpose, scope, applicability of the application, and the achievements of the group during development.

1.1 The Identified Problem and The Solution

MindfulStudent is a mental health application developed by our project group as part of the Design Project for Quartile 3 in the academic year 2023/24. While all members were aware that at one point the Bachelors in Technical Computer Science program at the University of Twente would lead us to develop a project to display our skills in programming, design, teamwork, and product development, MindfulStudent as an application transcends beyond just a rudimentary assessment of our skills in Computer Science. As students ourselves, we can relate to the ever-growing epidemic of Mental Health issues that have become increasingly and worryingly prevalent worldwide amongst the university-going population. Based on our own experiences as students, and those from our friends and peers, we are well aware of the difficulties that pupils face at this early stage of their life, it is a period of transition from a young child to an adult with responsibilities, dealing with the rather overwhelming abundance of knowledge presented in lectures, stressful periods of examinations, tense approaches of assignment deadlines, necessary fulfilment of household chores, commitment to healthy eating, a fit lifestyle, active social life and much more.

As discovered by Zivin, Eisenburg, Gollust, and Golberstein (2009) in their study published in the Journal of Effective Disorders, approximately half of the student participants reported having at least one mental health problem. Whilst this may seem concerning on its own, they also reported that an alarming 60% of such participants also reported having at least one mental health problem after two years. This would suggest that due to the delicate nature of the issue, mental health problems encountered by students may have a long-lasting effect on an individual's life, rather than getting better after more years at university or after graduating, as previously claimed. It can therefore be fairly justifiably argued that the stigma of mental health amongst students requires greater attention and an alternate approach to what is currently being offered. The demand for such services offered by universities has seen an upward trend (Brogia, Millings & Barkham, 2018), but not all educational institutes from across the world have the financial means and expertise available to offer all students access to such facilities, after all, there are an estimated amount of 254 million students worldwide.

What differentiates MindfulStudent from the current state-of-the-art is its attempt to capture technology as a means of fulfilling these requirements, an attempt to bridge the digital divide, and offer a single application that offers any student the ability to enhance their mental well-being. Whilst the ultimate goal of the application would be to perpetually address the various methods of enhancing student wellbeing, due to time, resource, and personal constraints the presented deliverable offers an Android Application that aims to provide our users (currently focused within the Netherlands) with a series of features that offer a holistic approach to enhance their mental well-being. Some of these features include sleep tracking, contact with mental health professionals, interactive breathing exercises, and personal journals amongst many more.

1.2 Purpose, Scope, and Applicability

1.2.1 Purpose

We developed the MindfulStudent app with the purpose of providing aid to students who deal with anxiety, stress, or other issues regarding their mental well-being throughout their studies. Being a student can be overwhelming especially when responsibilities stack up. Universities strive to provide support services to help students cope with such issues. However, it could be hard for some people to ask for help especially when the issue is regarding their mental health. It can be challenging for some people to openly ask for help. This is where the application comes into play: using its supportive features and user-friendly interface, it can provide a variety of options to guide and help students throughout their mental well-being journey.

1.2.2 Scope

The MindfulStudent app provides a variety of features with the purpose of helping students throughout their mental well-being journey. The features designed to assist with mental health issues are provided and described below:

- The breathing exercise feature specifically employs the 4-7-8 method to assist users in relaxing when they experience stress or anxiety.
- The chat feature enables users to share their issues with others, seek advice from fellow users and contact mental health professionals.
- The tasks feature helps users become more organised, aiming to reduce the stress caused by excessive worrying about their tasks.
- The Journal feature enables users to record daily journal entries, with the aim of motivating them to continue and allowing them to track their progress and contribute to their mental well-being journey.
- The sleep tracking feature allows people to keep track of their sleep time and quality, as it is an essential part of a person's mental health.

- The emergency contact feature offers essential contact information for use in emergencies, recognizing that some users may not be familiar with support contact details.

The main focus of the app is to help students deal with anxiety and stress caused by overwhelming schedules and responsibilities. The application also provides some contact information of professionals so that the student can find the help they need if necessary.

1.2.3 Applicability

The MindfulStudent app was developed with university students in mind. However, attention was paid to modularity and adjustability, and its features are not limited to be used by students. For this reason, the application could be used within any organisation to address mental health issues. While it was designed to be accessible to any age group, certain local laws may prevent the application from being used by people under a certain age limit due to the built-in chat feature. The storage problem due to the limited database usage can also be a challenge when the app is used in larger groups.

1.3 Achievements

During the development of the Mindful Student app, we as a team faced a lot of challenges and gained valuable experiences and insight as we overcame those challenges. The initial challenge for our team was to determine how to effectively identify mental health issues and provide appropriate solutions through app features. The complexity of mental health posed a significant challenge in designing features to address these issues. We as a team had to do lots of research to properly identify and address these issues.

As we progressed towards the development of the application, we encountered a problem related to the initial design of the application. We used Figma for the initial design, and when we transitioned to the development phase of the user interface, our team decided to use a built-in functionality where we could get the basics of the user interface as a code from Figma. While this functionality allowed us to convert basic designs into code, the lack of other format conversions led us to change our Android application into a Flutter application. Since our team had no prior experience with Flutter, we had to learn how to develop an app using it.

We encountered some minor issues during the implementation phase of our project. These issues were related to the lack of technical knowledge and we solved them by doing lots of research. One example issue we had was about real-time information extracting from the sleeping logs of Fitbit watch. One of the members of our team conducted research on possible ways to extract this information. After the research, he decided to do a static extraction instead of a real-time one so that the app would extract the information only when the dedicated button is pressed to

visualise. We have a lot of features in the app and each of them was developed by a different developer. Thus, we had some issues when it came to combining features, especially while working on the related UI pages. These issues were solved when we set a protocol for the development phase. The protocol was to assign a person who does all the merging in the project. Each member of the team created and worked on a new branch and when they were finished with their task, they would notify the assigned person to combine the work.

In addition to the technical experiences and insights that we have gained, we also dived into the philosophical aspects of the project, specifically on how emotions influence user motivation. Lastly, we accomplished all the planned tasks for this project by researching different planning strategies and applying them as we progressed throughout the project.

Chapter 2: Project Introduction

The following section will outline the various client and product details for the project. It will clearly define who we consider our clients to be, detail the application of our work on direct and indirect stakeholders, state the initial and final requirements of the project, use case diagrams to showcase the application functionalities as a whole, specify the software and hardware requirements for the basic operational setup and project planning with the methodology that is used throughout the project.

2.1 The Client

The Mindful Student app was requested by our team supervisor dr.ir. M.Zangiabady. She wanted our team to create an application to address mental health problems among students. At the start of the project, she requested some features that the app should contain and further explained how the application should serve students who are dealing with mental health problems.

2.2 Direct Stakeholders

2.2.1 Students

Students are the main target audience of our application. Since the primary goal of the application is to help students with their mental health. The students are the initial target however, the app is adjustable and can be used in any organisation for any group of people. There are no age restrictions for users so any age group would be a viable user. The age would not be a limiting factor for the availability. However, the database connection of the app has to be adjusted and modified if it were to be used on a larger scale due to the chat feature. Therefore, in short, the clients can be described as individuals seeking assistance for their mental health.

2.2.2 Mental Health Experts

Mental health experts are an important part of the app. Their contact information is provided within the app in case of a need for counselling such as extreme anxiety attacks.

2.2.3 Emergency Organizations

Emergency organisations will have their contact information provided within the app to allow users to reach out to them.

2.3 Indirect Stakeholders

2.2.1 Universities

Universities of course have an interest in seeing the mental health of their students improve. The application could supplement mental health programs as a way for students to exercise good practices in their own time. Therefore, universities could play a significant role in terms of advertising.

2.2.2 Organisations

Even though the intended purpose is for the app to be used among university students, it can be used within any organisation. Using the application could help people with their mental health and it can therefore indirectly increase the motivation of people in any organisation.

2.4 Requirements Specification

At the beginning of this project, we have outlined several functional requirements. All of these requirements are marked according to the MoSCoW system:

- (M) - Must have: essential features of the highest priority;
- (S) - Should have: important features, medium priority;
- (C) - Could have: desirable features, low priority;
- (W) - Won't have: optional features.

Our goal was to implement all Must and Should have features. Requirements that have made it into the final product have been underlined.

2.4.1 Account System

The application should feature an account system to identify the user.

1. (M) As a user, I should be able to create an account in the system using an email address and password of my choice.
2. (M) As a registered user, I should be able to log in to the system using my chosen username and password combination.
3. (M) As a registered user, I should be able to select what name and photograph my profile is associated with.
4. (W) As a registered user, I should be able to recover my account using my email address if I lose my password.
5. (C) As a registered user, I want the application to personally greet me on startup.

2.4.2 Sleep Improvement

The application should have a sleep tracking and improvement feature.

1. (M) As a user, I want the application to integrate with Fitbit or Google Fit to track my sleeping habits.
2. (S) As a user, I want to be able to view my sleeping patterns for each day over the past month.
3. (M) As a user, I want to receive a notification from the application to remind me that it is time to sleep.

2.4.3 Peer Communication

The application should have a chat feature that allows users of the application to communicate with each other.

1. (M) As a registered user, I want to be able to find my friends using the application and connect with them.
2. (S) As a registered user, I want to be notified when somebody wants to connect with me.
3. (M) As a registered user, I want to be able to send text messages to the people who I have connected with.
4. (M) As a registered user, I want to be able to receive text messages from people who I have connected with.
5. (M) As a registered user, I want incoming chat messages to show up in real time.
6. (S) As a registered user, I want to be notified when I have received a new text message.
7. (S) As a registered user, I want to be able to react to text messages with happy emojis, such as a heart or thumbs-up emoji.*

2.4.4 Mental Health Expert Communication

The application should provide a way to talk to a mental health expert.

1. (M) As a registered user, I want to be able to talk to a mental health expert through the app.
2. (M) As a mental health expert, I want users to be able to message me without having to connect with me first.
3. (W) As a mental health expert, I want to be able to share helpful text or audiovisual resources with users of the app.

2.4.5 Breathing Exercises

The application should have interactive breathing exercises to reduce stress.

1. (C) As a user, I want the app to help me do breathing exercises in order to reduce my stress levels.
2. (C) As a user, I want the breathing exercise to have at least one anime character for my enjoyment.

2.4.6 Digital Journal

The application shall have a personal digital journal.

1. (C) As a user, I want to be able to create a journal entry for a specific day.
2. (C) As a user, I want to be able to view my journal entries sorted by most recently created.
3. (C) As a user, I want nobody else to be able to view my journal entries.

2.4.7 Guided Sessions

The application shall offer guided study sessions.

1. (C) As a user, I want the guided study session to suggest breaks.
2. (C) As a user, I want the guided study session to be able to play music.
3. (C) As a user, I want the guided study session to suggest resources in order to reduce stress and fatigue levels.

2.4.8 Gamification

The application should nudge the user to improve their mental health through the process of gamification.

1. (C) As a mental health expert, I want registered users to be rewarded if they perform exercises in the app.
2. (C) As a mental health expert, I want registered users to be rewarded if they hit a certain daily exercise streak.
3. (C) As a mental health expert, I want the user to be reminded to do their daily exercises.
4. (C) As a mental health expert, I want the application to suggest users' well-being challenges (e.g. workouts per week, hikes, hours of sunshine, etc) in order to stimulate mentally positive activities.

2.4.9 Emergency Information

The application should provide the user with emergency information.

1. (M) As a mental health expert, I want the application to contain information about suicide prevention, police, fire services, ambulance, domestic violence helpline, child abuse helpline, sexual assault helpline, and legal aid services.

2.4.10 Book Tracking

The application shall provide users with the ability to track what books they've read.

1. (W) As a user, I want to be able to input books that I have read into my personal profile.
2. (W) As a user, I should be able to track what books I have read.
3. (W) As a user, I should be able to receive suggestions of books based on the books I have read.

2.4.11 Progress Tracking

The application should allow users to track their progress on any arbitrary task.

1. (S) As a user, I want to create a progress tracker that allows me to set up a certain numerical goal for a task.
2. (S) As a user, I want to set up the task according to a certain schedule.
3. (S) As a user, I want to be reminded by the app for the tasks that have a schedule attached to them.
4. (S) As a user, I want the progress tracker to motivate me to do my scheduled tasks.

2.4.12 Privacy and Security

The application should be private and secure.

1. (M) As a user, I want to be able to update any item of personal data stored at any time.
2. (M) As a user, I want to be able to delete all personal information related to myself.

2.5 Use Case Diagram

We have included a use case diagram of our final system below. It involves the various features of our application, as well as new and registered users as actors. Two interacting systems have also been included, namely Supabase (our backend) and Fitbit, which is the only sleep data provider we support at this time.

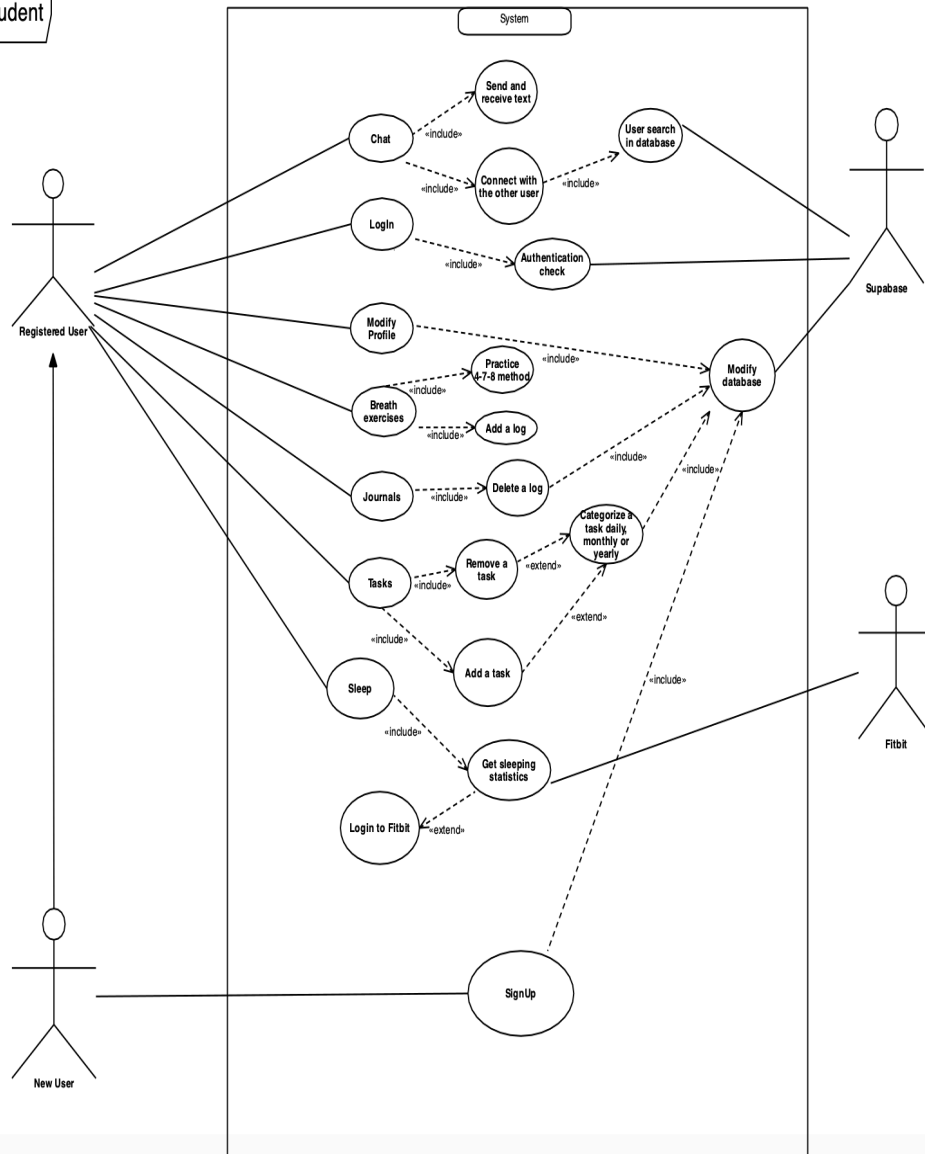


Figure 1: Use case diagram of the system.

2.6 Software and Hardware Requirements

In cooperation with the client, we identified two main hardware requirements:

1. The application should be compatible with Android 10 and higher.
2. The application should support any arbitrary screen size of at least 960x640px.

While software requirements were not explicitly defined, we did set some internal goals to align with values that we deemed to be important, such as privacy and maintainability. For example, when choosing a back-end architecture, we opted for an open-source, self-hostable solution, which increases privacy and ease of maintainability. Also, while support for iOS was not set out

as a requirement, one of the reasons we chose our current front-end framework was to make it easier to port the app to Apple devices in the future. The sleep tracker integration was also built with extensibility in mind, in order to make it easy to add additional data providers in the future. All of these points will be discussed in more detail further on in this document.

2.7 Teamwork and Planning

Throughout the project, we have consistently utilised the agile and scrum methodologies to develop the application. This consisted of a shared, internal Trello board where we kept track of our sprint progress, backlogs, and task assignments. Each and every task was also supplied with a start and end date, to make it easier to track how long people have been working on a certain task. This allowed us to more efficiently dedicate our resources and check up on or offer help to team members if needed.

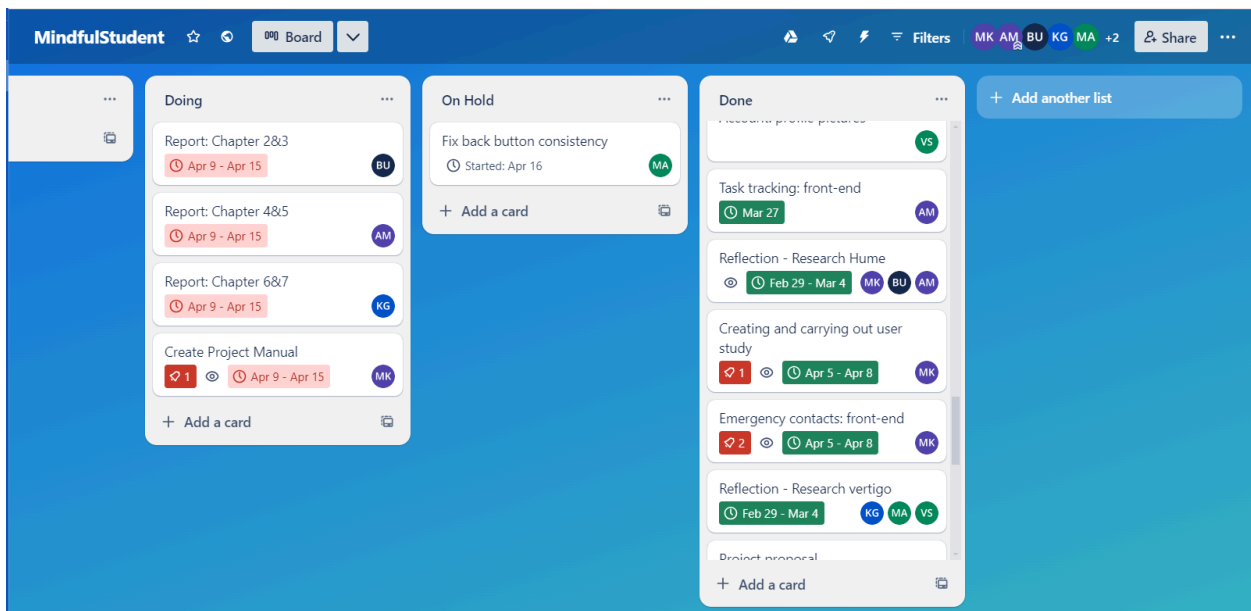


Figure 2: Our team's internal Trello board.

Besides this, we also held daily meetings where we discussed our progress and divided the remaining tasks. Unlike what typically happens during a scrum, these were usually planned after lunch: we noticed that giving people some time to work on their tasks in the morning resulted in more effective stand-ups because the problems were still fresh. Sometimes, when we held meetings in the mornings people would explain the problems they faced the day before but had gained additional insights without having had the time to work those out yet. This resulted in them being less likely to ask for help and others being less likely to offer it, reducing the effectiveness of the stand-up.

We divided the project into 4 sprints. For each sprint, we planned, designed, developed, and tested the dedicated tasks. At the end of each sprint, we presented the progress to the supervisor and asked for feedback. Based on the feedback received, we adjusted the planning for the following sprint. A rough overview of our final sprint planning is shown in the table below.

Sprint 1 (12/02/2024 - 01/03/2024)	Sprint 2 (04/03/2024 - 15/03/2024)	Sprint 3 (18/03/2024 - 29/03/2024)	Sprint 4 (01/04/2024 - 14/03/2024)
Initial client meeting	Finalise application design	Push notifications	Communication with health experts
Project proposal agreement	Determine technology stack	Chat front-end	User / Security testing
Requirements elicitation	Application login/registration system	Sleep tracking implementation	Write project a report and a manual
Application design (including colour schemes etc.)	Chat back-end	Emergency contacts page	Finish missing features (Tasks page and Progress bar)
Initial technical and literary investigation	Fitbit integration & Sleep tracking	Sleep tracking statistics	Make a poster and prepare a presentation

Figure 3: Final sprint planning

During the project, people have worked on several parts of the system or the report. The individual contributions for each team member have been mentioned below.

Batuhan Usta: Implementation of the notification feature for the backend and the frontend, UI and functionality implementation of the breathing exercise, philosophy report for the project, and the section 1 - 3 of the final project report.

Aarav Ashish Mehrishi: Converted figma designs to flutter for login flow, home screen, profile, task tracking (with backend), chat and carried out and reported system tests for the application.

Veselin Shterev: Mainly working on the front-end of the app, initial Figma design ,Journal page front-end and back-end, chapter 4 of the report

Mike Almelo: Back-end architecture, integration of front-end with the back-end, Fitbit integration. Wrote mostly technical documentation for the report.

Umar Kashif: Implemented the front end for the Emergency Contact Page, designed, carried out and reported the findings of the User Survey & Usability Testing and created the Project Manual.

Kristian Guenov: Provided support for the design model on Figma, researched back-end to front-end connection for chat features(however help was requested due to technical issues), contributed to the final version of the report.

Chapter 3: User Interface Design

In this chapter, we will discuss our UI design philosophy and the values we deemed important while designing the application. We will discuss the various design iterations we have gone through, outline the pros and cons of our intermediate designs, and explain how and why we have come to our final design.

3.1 Initial Design

The initial UI of the application was designed using Figma. The application provided us with a lot of features to easily build the initial UI pages. It was an important aspect since the purpose of the initial UI design was to get feedback from the client as soon as possible. We received positive feedback along with some minor change requests. As a result, we determined that there were no major changes necessary to the overall design of the application. The UI design initially had 16 different pages (see Appendix A). These pages were serving the purpose of the features mentioned in the requirements section of the report. Some of which were only serving the purpose of informing users. These pages were not functional and had no critical purpose to the application therefore, they were not used in the final design (see Appendix C).

3.2 Final Design

The final design was structured from the initial design. The pages created in the initial design phase were converted into code. These codes did not visually match the drafts we created using Figma. Therefore, we used the background code provided by the conversion and implemented the rest ourselves. This issue set us back on our schedule and we had to compromise on some of the optional requirements that we set initially. The final design of the application contains all the required UI pages with some optional ones (see Appendix B).

3.3 Design Philosophy

During this project, we have paid extra attention to the design of our application. Given our target group, we have tried to make our application look as peaceful as possible. This philosophy requires the cooperation of several tactics: sticking to a consistent colour scheme, re-using elements throughout the design, and keeping the properties of all elements, such as roundness and shadows similar to each other.

Other than consistency, the general theme of our application is also important and consists of more than the previously mentioned techniques. The colours we used were specifically chosen to induce a peaceful mood: green is widely seen as a peaceful and relaxing colour, which is why the UI consists of a light theme with mostly green-derived pastel colours (McLachlan & Leng,

2020). There are a few exceptions to this: the sleep tracking overview also uses light blue and yellowish green to differentiate between sections of the UI, for example. However, these were specifically chosen to complement the green of the application itself, and we made sure not to make them “pop” too much.

Finally, we designed our application to be mobile-first. Most students carry their devices with them at all times, so we felt that a mobile-first experience would greatly enhance the accessibility of our application. Additionally, students often have busy schedules and this approach would allow them to access mental health resources on-the-go, providing a better fit compared to desktop or web interfaces.

For the design, this approach means that a navigation bar can be found at the bottom of the screen for easy single-handed use, and the application integrates with the system back button to follow the user’s preferences. Screens have been designed to fit on a single page but will overflow vertically if required, which is more convenient on mobile devices. Additionally, the top header bar integrates with the device’s status bar and its colours will stretch all the way to the top of the screen, rather than showing a black bar or pushing the status bar out entirely. This provides an immersive experience while respecting the user’s preferences and the functionality of the device.

3.4 Design improvements

The design process for the application comprised two main phases: the initial and the final design. The objective of the initial design was to create a draft for presentation to our client. However, due to a delay in the meeting schedule, we found ourselves with extra time to work on the initial design. This extension along with extra time allowed us to develop a more advanced initial design. Following the meeting with the client, we received positive feedback along with some minor change requests. As a result, we determined that there was no need to further improve the initial design in Figma, as these changes could be incorporated during the conversion process.

The conversion process did not go as smoothly as we planned. The Figma application provides an extension to convert the designs into codes. However, the conversion was not accurate as we tested them in our Flutter application. We addressed this issue by taking the background from the converted design and then proceeded to implement the remaining components ourselves. This solution set us back on our schedule and we compromised on some optional feature pages. Nevertheless, we were able to achieve a final design that received approval and was well-received by our client.

Chapter 4: Ethical and Security Considerations

4.1 Ethics

Considering the ethical implications of our app is essential to our effort to improve students' mental health. This guarantees that the app complies with moral principles that safeguard and honour users, particularly in light of the delicate nature of mental health disorders.

First and most important for such apps is transparency and honesty. We made our app's goals, features, and restrictions very apparent. We believe that users should be made aware that the app is a tool to make getting professional mental health care easier and enjoyable, rather than a replacement for it. Users also have complete control over their data, including the freedom to access, alter, and delete it on the spot. Since we are aware of how serious mental health problems may be, the app also provides essential features like contact details and direct access to mental health crisis help lines. This guarantees that users will get expert assistance right away when needed, making the app a dependable partner in taking care of their mental health.

The app is dedicated to upholding the accuracy and reliability of its content by strictly adhering to evidence-based methodologies. The app's exercises and guidance for mental health are all based on the most recent research and approved best practices. Additionally, the app benefits from frequent updates and evaluations from experts in the field, ensuring that all information offered is current as well as secure and effective. This expert participation upholds the app's credibility and guarantees that it is a reliable resource for students managing mental health issues.

4.2 Security

Security is essential to our database architecture and operation, as the app manages sensitive personal data such as journal entries, sleep patterns, and mental health records. This was one of the major factors for deciding to go with Supabase as our backend provider: Supabase is a proven product that is used by companies and organisations worldwide, and by leveraging its existing security features, we can be more confident about the security of our own application as well. The platform itself is SOC 2 and HIPAA compliant, which requires regular auditing by an external company. All of Supabase's components are open source and therefore independently auditable by anyone who wishes to do so.

For authentication we use Supabase's built-in GoTrue authentication system. To our knowledge there have been no major security incidents in this product, and its widespread usage reinforces our beliefs that this system is stable and secure. Supabase uses this module for their own platform as well and have used it to pass independent security audits as mentioned before. Secure salted password hashing with Bcrypt as well as further protections against brute-force attacks,

such as strict password restrictions and cross-referencing against a known database of hashed passwords, are the foundations of this authentication system.

Authorization is managed directly in PostgreSQL using row-level security (RLS) and column-level security (CLS). Both of these features are widely used and integrated directly into Postgres, which is an incredibly popular open-source DBMS used for all kinds of databases. Our backend architecture allows us to execute many tasks directly in PostgreSQL using triggers and `pg_cron` jobs, leveraging the world-class security and dependability of the DBMS itself. Additionally, by not exposing these operations and their involved data to any outside processes, we can greatly reduce our attack surface to the outside world.

While our application essentially directly accesses the remote database for its operations, this does go through a proxy layer called PostgREST. PostgREST heavily relies on parameterization which effectively nullifies the probability of an SQL injection attack.

Of course, our backend is only as secure as we configure it to be. RLS and CLS rules need to be configured manually, and misconfiguration could allow people to perform actions on the backend which they are not allowed to do, such as querying other users' data. To prevent this, we have adopted a bottom-up approach where initially, any and all users are disallowed access to new database tables. We then progressively add new declarative policies for each intended feature, such as "Allow users to access their own journal entries." These policies stack and as long as one of them matches, the user is allowed to execute the query. By keeping the policies small we ensure that they are easily testable for mistakes.

4.3 Privacy

Privacy was an incredibly important requirement during the development of our application. We have ensured that any piece of information uploaded by the user will always be modifiable and deletable. Additionally, deletion of their account will remove all the information that has been stored about them, with two notable exceptions: chat messages and access logs. After discussion with our client we decided that chat messages would instead be detached from their user account, to allow them to be kept by their peers for future reference. They are therefore anonymized. Access logs containing the source IP address, HTTP method, host,path and response code are currently stored indefinitely on our development server, but this parameter is configurable for real-world deployment.

Despite this, in its current form the application is not GDPR compliant. The main reason for this is its management: due to the scope of our project, there is currently no contact information for users to access, or a privacy policy when signing up for an account. Instead, we designed the

application to be “gdpr-ready”: to our knowledge, the actual, technical implementation of the product complies with privacy regulations to our best extent.

This entire approach to security and privacy guarantees that user data on MindfulStudent is held to the highest standards, offering a dependable and secure setting for handling individual mental health data.

Chapter 5: Implementation

5.1 Technology Stack

For our technology stack, we initially wanted to use a regular HTTP server, such as Spring. However, after considering our heavy focus on the user experience and therefore the front-end of our application, we started looking into alternative solutions that are quicker to get started with, to allow ourselves to shift more of our focus to the application itself.

At this point, Firebase was the most logical choice. It is a mature platform operated by Google which provides first-class support to mobile applications. Built-in authentication and being able to directly access the database from our mobile application (given some security constraints) was a huge advantage, because these are some of the most time-intensive tasks when setting up your own back-end. However, storing all user data on Google servers and the locked-down approach of Firebase together with its pricing model repelled us, so we decided to look into alternative options.

We briefly looked into Appwrite, which claims to be an open-source Firebase alternative. Appwrite is still rather new however, and does not allow for the fine-grained access controls we need. Therefore this option was also discarded.

As a last resort, we tested Supabase. Supabase is a platform that tries to mimic Firebase in terms of features, but is completely built on open-source technologies and is fully self-hostable. Almost the entire platform is built on PostgreSQL, making it very easy to get started with and very reliable in our experience. It features built-in authentication, and access controls are implemented using Postgres's built-in Row-Level Security (RLS) and Column Privileges. These privileges can be specified per database operation. This allows us to, for example, specify that any registered user may insert any message into the messages table, as long as they specify themselves as the sender, and there is a connection (friendship) between them and the recipient in the connections table. An additional bonus is that these policies are completely declarative and self-contained; this makes it harder to introduce bugs and easier to test.

Sometimes, additional back-end logic is needed, for example to handle the sleep tracking login flow. For this purpose, Supabase features Edge Functions which allow us to write a more advanced HTTP server in plain TypeScript. These can also leverage Supabase's authentication system by preventing unauthorised requests from reaching the Edge Function in the first place, which enhances security.

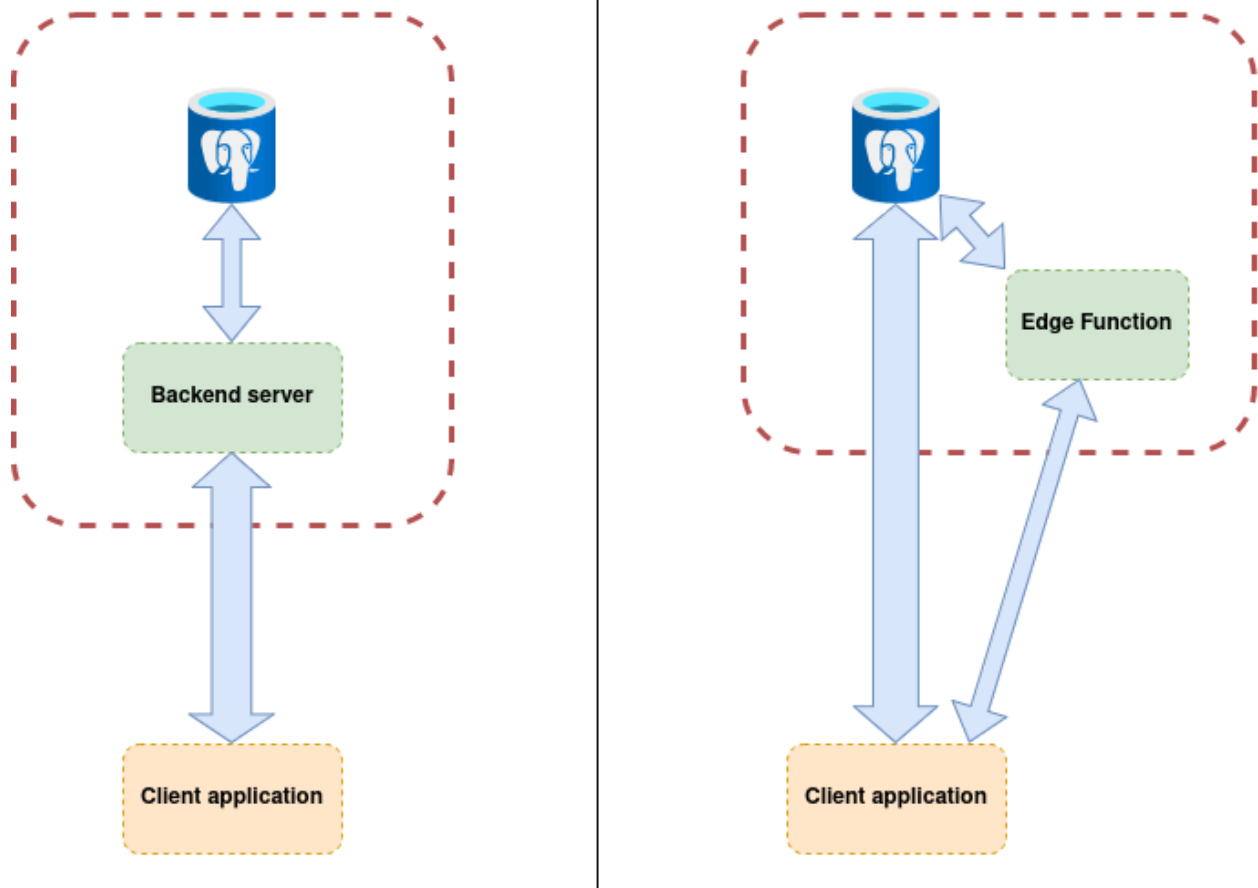


Figure 4: Graphical representation of a typical back-end infrastructure (left) and our Supabase-based infrastructure (right). The size of the arrows indicates the amount of communication happening between the modules.

For our front-end, we initially started with a plain Java app, but switched to Flutter shortly afterwards. Flutter has great community support and allows us to write code for a single application that will, in theory, work on multiple different platforms. While an iOS app was out of scope for our project, this does ensure that there exists an easy way forward in case the scope gets expanded in the future.

5.2 Implementation Details

While building the application, we adapted a parallel development approach. What we mean by this is that we developed the front- and back-end independently from each other. In the front-end we developed a very thin adapter which acts as the connection between the UI and the back-end database; this way, when the back-end team made a change they only had to update the code in this adapter. The adapter then provides a stable API that can be used by the UI team. This

worked especially well in our case since most people would be working in either one of the “implementation teams,” but not both at the same time.

5.2.1 Authentication

For authentication, we use Supabase’s built-in GoTrue module. GoTrue uses bcrypt to store password hashes, which are of course salted for extra security. Additionally, we have set up extra password requirements to combat brute-force attacks: passwords must be at least 8 characters long, and every password is checked against the HIBP Passwords API, to prevent people from using passwords that have previously been involved in a breach. The authentication flow is performed using the aforementioned adapter on top of the Supabase SDK, which provides us with common actions such as logging in and signing up.

Sessions are kept track of using JSON Web Tokens (JWTs). This is, simply said, a JSON object containing data about the logged-in user and an expiration date of 24 hours. It is stored on the client side but cryptographically signed using a secret key on the server, making it impossible to modify without triggering authentication failures. After expiry, the client will attempt to “refresh” the connection. The advantage of JWTs is that they can be verified by checking a cryptographic signature instead of performing a database lookup, which is much more performant and can happen in a distributed manner. This positively impacts the scalability of our application, especially since the authentication endpoints are called quite often.

5.2.2 Database

Thanks to Supabase, 80% of our backend architecture consists of nothing more than a PostgreSQL database. From the client we connect (almost) directly to the database using PostgREST, which is a REST and WebSockets interface for Postgres and directly integrates with GoTrue. This allows us to execute regular database operations, but also provides us with a stream in order to listen to changes such as incoming messages. By securing our database tables using RLS and Column Privileges as mentioned before, we can provide secure, direct database access from the client without needing to build our own REST API.

Since most of our backend architecture consists of just a database, this is also where most of the performance can be gained. In order to increase this performance, we make heavy use of database indexes on commonly-selected columns. Additionally, by normalising our data as much as possible, destructuring it into separate tables and carefully choosing our primary keys, we are able to query data more efficiently.

Still, a lot of performance is lost due to query delays. In a typical backend architecture, sending multiple consecutive queries to the database is not that much of a problem, since the latency between the backend server and database is generally only a few milliseconds. However, in our

case this practice comes with heavy performance penalties due to the large RTT (round-trip-time) to our database. We were unfortunately not able to completely remedy this; PostgREST queries are structured slightly differently compared to regular SQL, and we were not able to completely figure this out in time. To remedy this, we do try to batch up queries where possible to send them at the same time, which effectively runs all of these queries in parallel.

Another advantage of heavily relying on foreign keys is operation cascading. When we delete a user's account, the row deletion cascades to all tables that depend on that account, effectively removing all of that user's data from our database. This ensures that all data is actually deleted and we do not accidentally miss anything. One exception to this is the message data however: after discussing with our client, she preferred that messages would not be deleted when someone deletes their account, to allow their peers to reference historical chat messages. This data is instead anonymized by nulling the message author, which is another cascade option. People are still able to manually delete their own messages at any time.

Some more complicated tasks, such as generating notifications on specific actions and cleaning these notifications up after a while, are also directly managed in Postgres. For example, to generate push notifications we use triggers in order to insert a row into the notifications table when needed. Using another trigger on the notifications table, this insertion generates a call to an Edge Function which then delivers the notification to the user's device.

Similarly, to prevent old notifications from lingering around, the cleanup of this table is done using [pg_cron](#). This allows us to delete old notifications on a regular schedule, in our case 7 days.

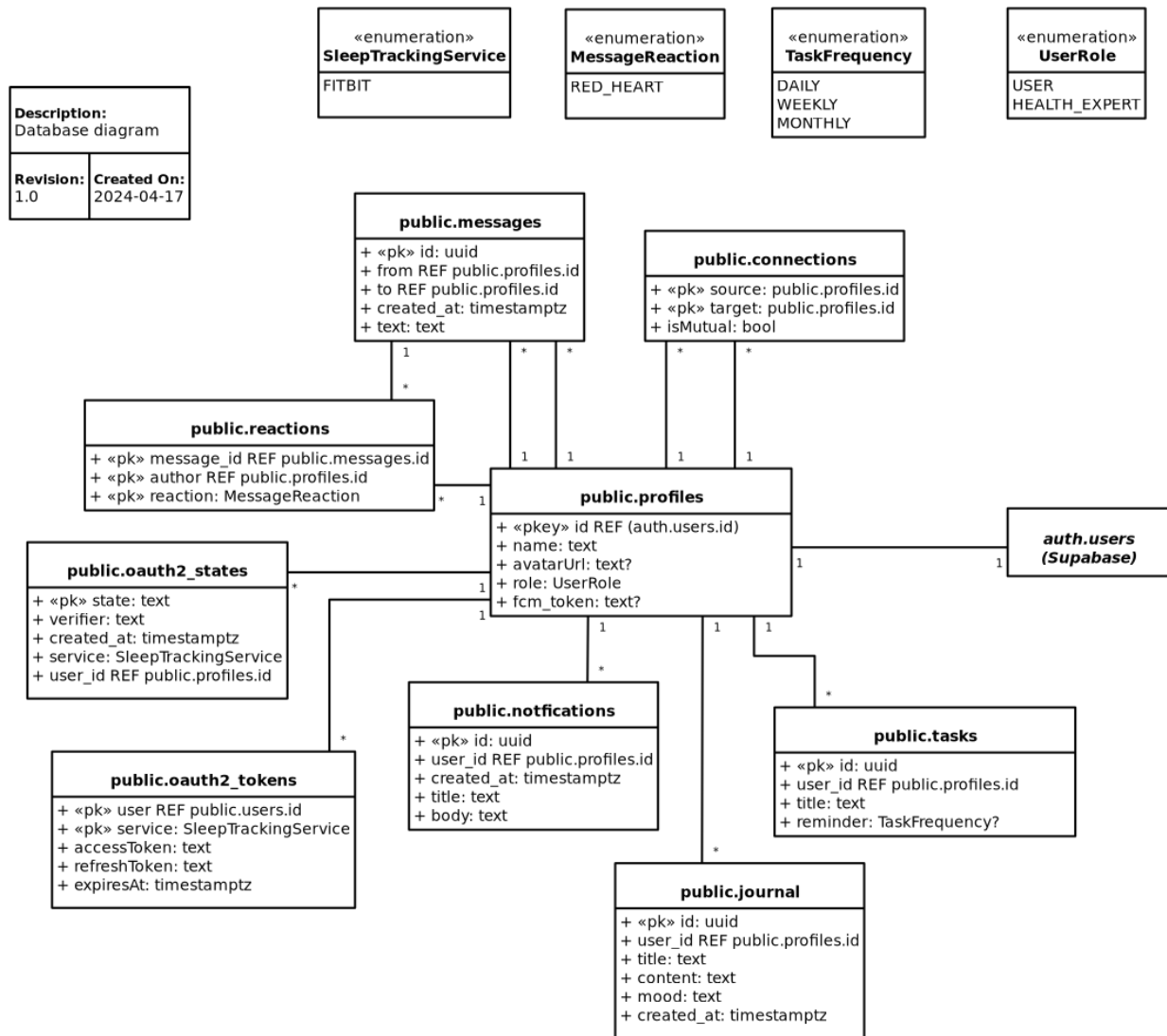


Figure 5: Diagram of our database structure.

5.2.3 Supabase Edge Functions

Supabase has a feature called Edge Functions. As the name suggests, edge functions are “functions” that are able to operate on the edge, independently from the main backend infrastructure. They consist of small web servers using TypeScript that execute on the Deno runtime, and have administrator access to the database. While the majority of functionality of our application can be modelled using PostgreSQL alone, some features, such as sleep tracking and account deletion, require extra logic and are therefore implemented as an edge function. Some downsides of using edge functions is that they take longer to build, and it is easier to make

authentication mistakes. While implementing security policies directly in the database is done declaratively (user “x” can view / insert these rows), edge functions follow the typical procedural programming model, and database communication needs to be done manually.

Luckily for us, these functions can also profit off of GoTrue’s authentication mechanism, and by default, unauthenticated requests are blocked from reaching the edge function in the first place. Since authentication uses a simple JWT and we know that that JWT must be valid (the request wouldn’t have been received otherwise), we are able to parse the user’s account details from this JWT relatively easily.

5.2.4 Sleep Tracking

Our application features sleep tracking and insights by integrating with external data providers. While we currently only have support for integrating with Fitbit, both the front-end and the back-end have been designed to support multiple other providers as well, such as Google Fit. Connecting and maintaining a connection to a third-party data provider service is usually done using the OAuth2 standard. This is a generic authentication standard, and in our case we are using it to connect a user’s Fitbit account to their MindfulStudent account. This connection process consists of 3 parts, namely initiation of the connection, receiving and processing the connection confirmation, and maintaining a usable access token to retrieve data from the 3rd party service. The exact process has been outlined in the diagram below.

Description: OAuth2 connection process for sleep data providers	
Revision: 1.0	Created On: 2024-04-17

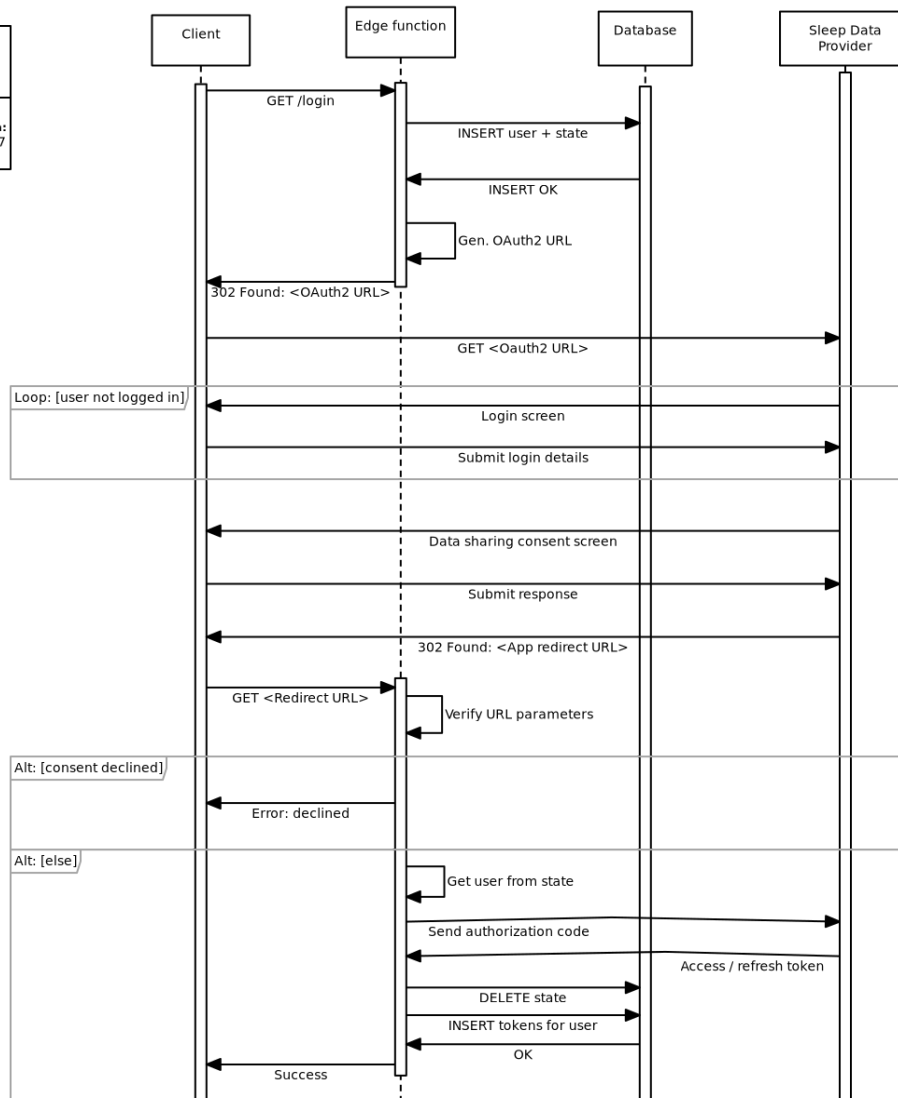


Figure 6: Sequence Diagram detailing the connection process for sleep data providers.

In order to initiate the connection, we generate a URL with specific parameters, as indicated by the service. One of these parameters is a state: a cryptographically random token which corresponds to the specific login request. We save this token in our database along with the user who requested the connection attempt; we will need this information later. After this, we present the generated URL to the user in a browser window. The URL points towards a specific webpage of the sleep data provider, and they will make sure that the user is logged into the provider's account. They also make sure to ask the user whether they want to share their account data with us.

After the user either confirms or declines, they are redirected to another URL which we registered with the provider in advance. There is an edge function running on this URL that is ready to receive the connection response. If the user accepts the connection, the URL will contain our random state token as well as an authorization code. The authorization code is exchanged for an access and refresh token by communicating directly with the 3rd party service, and the user is identified by retrieving the previously stored state token from our database. At this point, the account is “connected,” and we store the access and refresh tokens in our database, connected to the user we retrieved from the state.

Note how in the diagram, the lifetime of our edge function is rather short: edge functions only run for as long as they need to in order to save resources. They wake up when an HTTP request comes in and shut down once they have generated a response. Because we require a state in order to keep track of which account should be connected (as well as for security purposes), we have a database table that acts as temporary state storage.

After the login process, we are able to use the access token to fetch sleep data from the user by directly contacting the connected service. However, due to security requirements, the access token can expire, and it will need to be refreshed using the refresh token. In this process the refresh token is exchanged for two new access and refresh tokens, which are to be stored in the database again. This process happens completely transparently from the user. See the diagram below for an overview; in the interest of simplicity, database transactions have not been modelled.

Note how the above scheme does not involve actual storage of the sleep data. This makes our approach more privacy-friendly since users are able to manage their personal data directly at their provider instead. Changes to the provider’s data will then be reflected in the app in real-time.

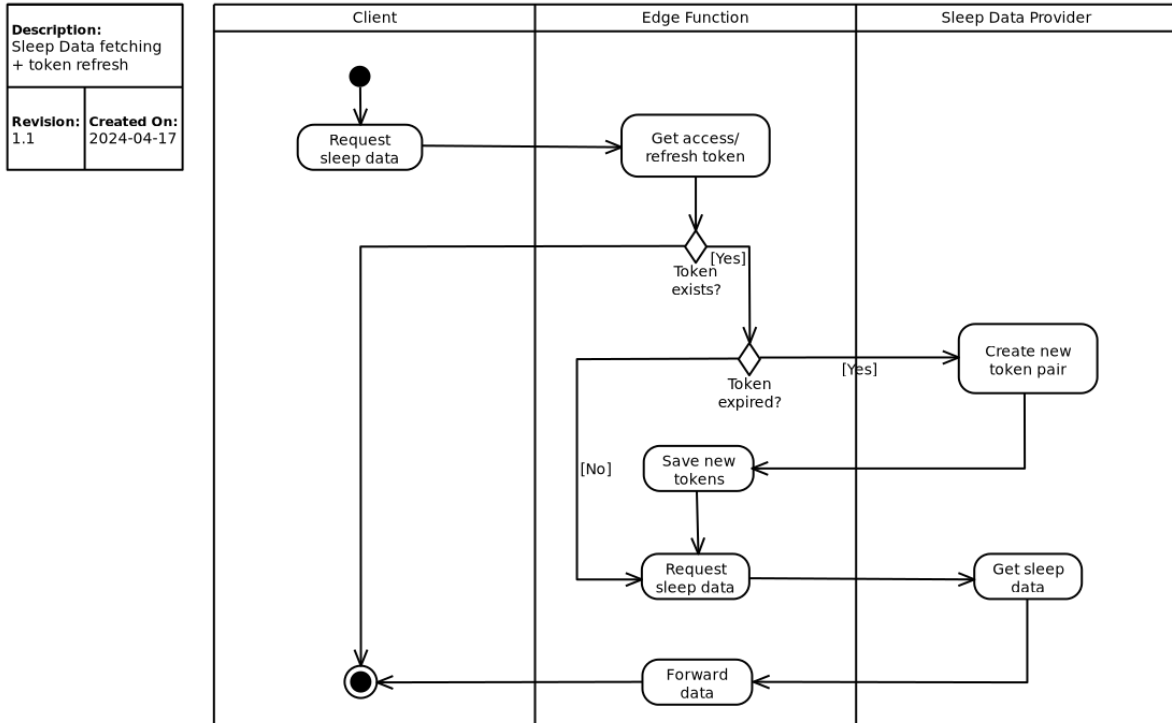


Figure 7: Activity diagram of the sleep data fetching process.

5.2.5 Push Notifications

In order to deliver push notifications to users’ devices, we are still required to use Firebase. Google has complete control over the push notification system on Android, forcing us to use their service for this feature.

In order to receive push notifications, every time the application starts up we use the Firebase SDK to retrieve a special token from Google. The application will upload this token to our database, where it will be stored to be used later. The token can be used from the back end to send a message to Google, and Google will ensure that the message is delivered to the user’s device as a notification.

As briefly mentioned before, when anything notification-worthy happens in the backend, we use a PostgreSQL trigger to insert a corresponding notification entry in a separate database table. Because sending a notification to Google is a slightly more involved process, this needs to be done using an edge function. For this reason, we then have another trigger set up for the notifications table that activates when a row is inserted, which will execute said function. The edge function retrieves the inserted notification from the table and uses pre-saved credentials to

communicate with Google and send the actual notification. Google then ensures that said notification is delivered to the end user.

sd Notification System

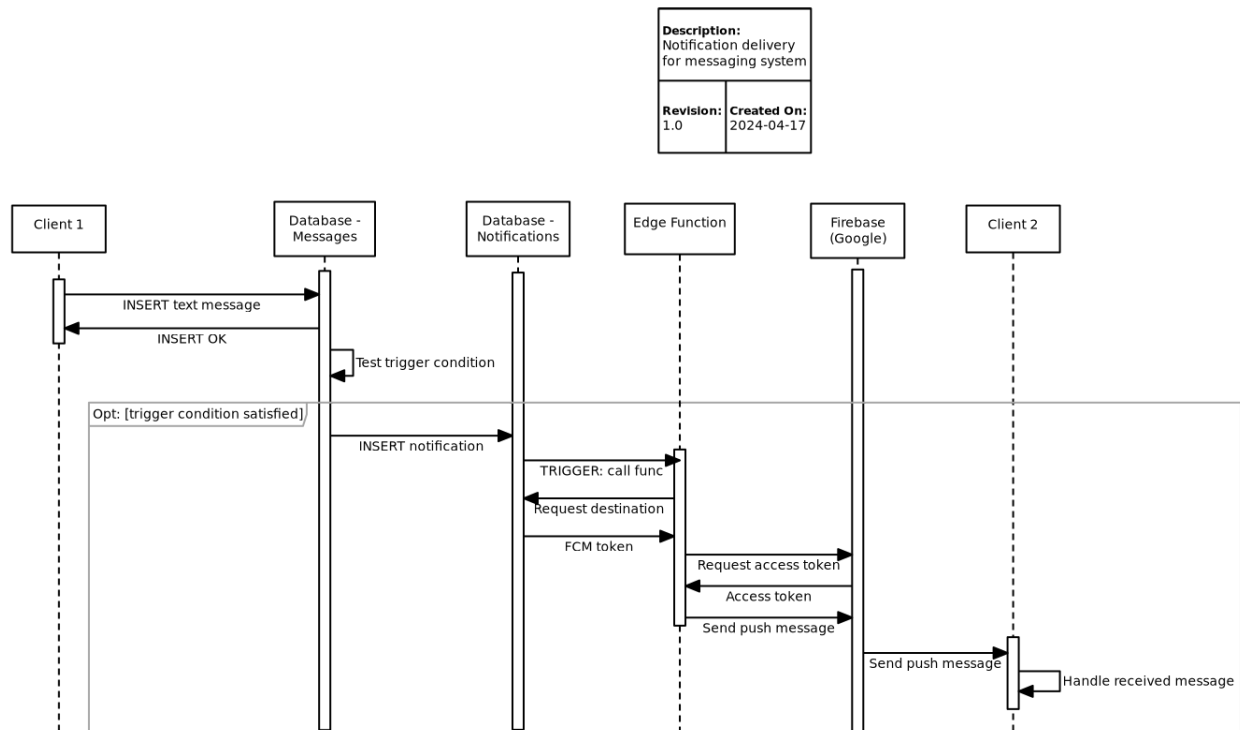


Figure 8: Example of the notification delivery system. Client 1 sends a text message to Client 2.

In some cases, we do not want the received notification to be displayed, for example when the app is already open. Therefore we register a message callback using the Firebase SDK to intercept the incoming message, and post it as a little pop-up banner in the app instead. When the app is not open the callback isn't registered either, and Google will automatically show a pop-up notification. See Figure 9 for more details on how this works.


```

static Future<void> init() async {
  // If Firebase tells us the token changed, update it in the DB
  FirebaseMessaging.instance.onTokenRefresh.listen((fcmToken) async {
    await Firebase._updateToken(token: fcmToken);
  });

  // Register message listener
  FirebaseMessaging.onMessage.listen((payload) {
    // We received a push message while the app was open!
    final notification = payload.notification;
    log("Message received: ${notification?.title}");

    // Show a snackbar at the bottom of the screen
    scaffoldMessengerKey.currentState?.showSnackBar(
      SnackBar(
        content: Text('${notification?.title}:
${notification?.body}'),
      ),
    );
  });

  // Required for Firebase init
  await FirebaseMessaging.instance.requestPermission();
  await FirebaseMessaging.instance.getAPNSToken();

  // Update token immediately on app startup
  await Firebase._updateToken();
}

```

Figure 9: Our Firebase initialization code. It registers a listener when the app is open to intercept a push message and show it in-app instead of as a notification.

5.2.6 Flutter Data Integration

Using the general architecture as described above, we are able to successfully exchange data between our application and the backend. But how do we show the correct information to the user? One way would be to manually update the UI once we receive updated data, and while this would work perfectly fine, it is prone to errors and more difficult to get right due the large amount of edge cases.

Instead, we use Flutter's built-in Provider-Consumer system. Our app consists of a nested widget

tree that makes up the application, and all the way at the top of the tree, we register multiple so-called providers, usually one for each data point. A provider consists of a relatively simple class that has only one task: keeping track of the data it is responsible for, and ensuring it is up-to-date.

```
class UserProfileProvider with ChangeNotifier {
  // Private field to store user profile
  Profile? _userProfile;

  // Public getter for profile
  Profile? get userProfile => _userProfile;

  // Set current user profile and notifies consumers to re-build
  void setProfile(Profile? newProfile) {
    _userProfile = newProfile;
    notifyListeners();
  }

  // Update saved user profile by fetching current account details
  Future<void> updateProfile() async {
    final profile = await Auth.getProfile();
    setProfile(profile);
  }
}
```

Figure 10: Code snippet for our user data provider. By calling `notifyListeners()`, we let all subscribers know that the data has changed.

```
@override
Widget build(BuildContext context) {
  return MultiProvider(
    providers: [
      ChangeNotifierProvider.value(value: profileProvider),
      ...
    ],
    child: ...
  );
}
```

Figure 11: The providers are registered at the root of the app's widget tree. This way, we can register consumers anywhere in our app.

The above structure allows us to register Consumer widgets anywhere in our application, which are essentially a declaration that its child widgets are dependent on the data of the provider it is subscribed to. When a provider decides to update its data, all consumers are notified and will be instructed to rebuild their widget tree. Flutter will then figure out which parts of the tree actually need to be updated, in order to improve efficiency. This system allows us to efficiently update data in the app and more easily embrace lazy loading, which makes it start up faster and feel much snappier in general.

```
Consumer<UserProfileProvider>(
  builder: (context, profileProvider, child) {
    final Profile? userProfile = profileProvider.userProfile;

    return ...
  },
)
```

Figure 12: Example of a consumer subscribed to the user profile provider from Figure 10. When the provider updates its data, the builder function gets executed and the widget tree will be rebuilt, ensuring that the data shown to the user is always up-to-date.

Chapter 6: Testing

Testing is a key component of any developed Computer System and covers a wide range of purposes, from ensuring all client expectations are met, the system operates as the developer intended to do so, and evaluating the system's performance under stress amongst many more. While there exist several forms of testing, the primary focus for the Mindful Student Application's testing stage was to ensure that the product meets the requirements of the target user. To do so, we conducted user testing in the form of surveys to gather the requirements of students in the context of their mental health, as well as going through the application with some users to ensure that the usability of the app was as intended. Furthermore, we conducted System Tests where the requirements of the application were tested to ensure that the behaviour of the final app was as intended. Due to time constraints, we were unable to properly conduct any form of automated testing, since setting up such an environment for such tests to be conducted was too complex and time-consuming and our focus was primarily on testing the usability of the app, since that provides greater assurance of its ability to meet our goals. However, had we had a few more weeks of development time, we would have conducted Widget Tests to ensure that widgets behave as expected, and looked into security testing.

6.1 User Testing

There were two key forms of user testing that were conducted in the testing phase of the application. These are listed below:

- **User Survey:** A wide range of fellow students were asked to fill in a questionnaire in which they were asked questions related to their demographic details, sleeping patterns, general mental health questions and their preferences about the application.
- **Usability Testing:** 3 students from the University of Twente were picked to conduct a more thorough usability test on the application. They were provided with an early release version of the application and asked to interact with the application and test the various features that are provided. They were then asked to provide feedback on the application, their likes and dislikes, what (if any) confusion they faced, what improvements they would suggest amongst much more.

The set-up of these tests, the results and their interpretations can be found in the User Survey and Usability Study sections in the appendix.

6.2 System Testing

System tests involve testing on a complete integrated application rather than testing each component individually.

How to read the instructions:

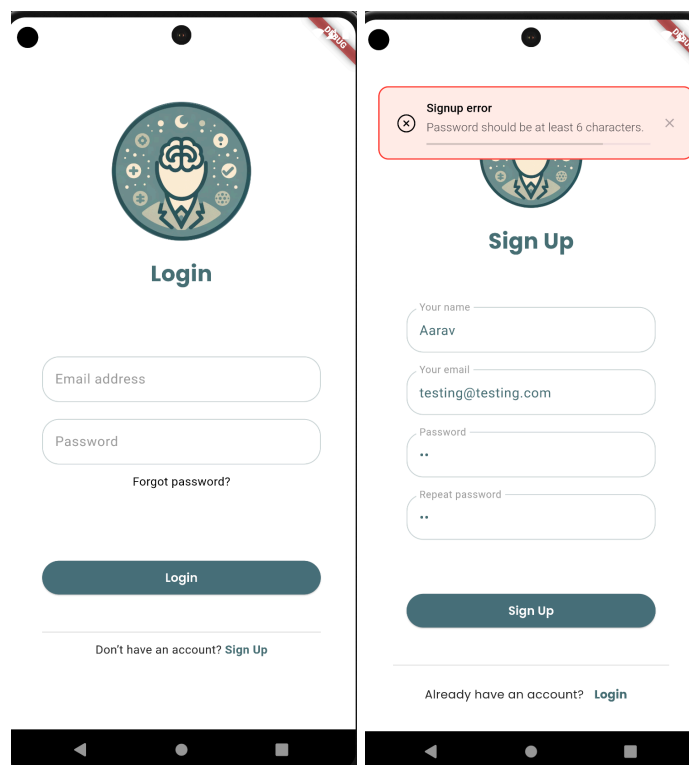
- In order to reduce redundancy in the instructions for the tests, setting up and running the application is written separately.
- When a step is written in **bold**, a condition needs to be checked.

Setup and Run the Application:

1. Install the application on the target device by using the .apk file provided by our team.
2. If prompted, allow installation from unknown sources.
3. Press “install” to install the application to the device.
4. MindfulStudent should now be visible on the home screen.

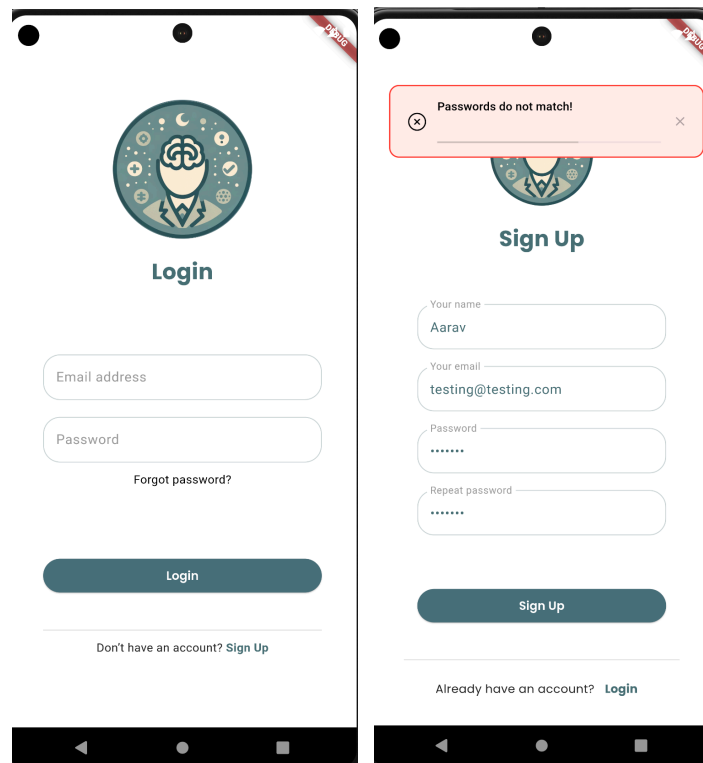
Password Length Requirement Test :

1. Setup and Run the application
2. On the login page, click Sign Up
3. Enter your credentials and a password which is less than 6 characters
4. Click the Sign Up button
5. **Check that a notification for a signup error is received, since the password should be at least 6 characters and the user is not allowed to sign up.**



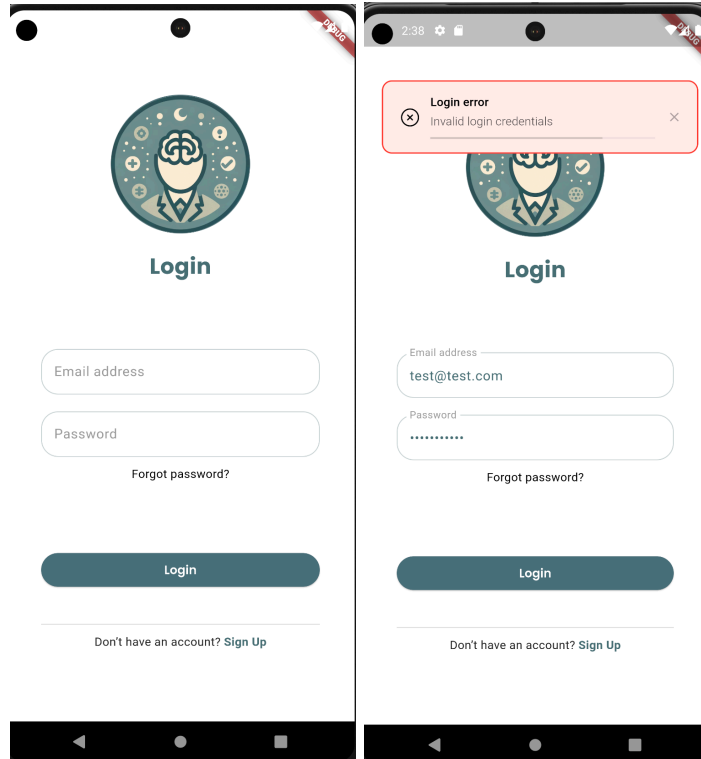
Password Mismatch Test:

1. Setup and Run the application
2. On the login page, click Sign Up
3. Enter your credentials and make sure the password and the confirm password fields are not the same.
4. **Check that a notification for Passwords that do not match is received and the user is not able to sign up.**



Invalid Login Credentials Test:

1. Setup and run the application
2. On the login page, enter incorrect credentials
3. Click log in
4. **Check that a notification for Invalid login credentials is received and the user is not allowed to log in.**



6.3 Modifications and Improvements

Kindly refer to the Main Takeaways and Interpretations sections from Appendices D & E which offer great insight into how the results of the user survey and usability testing changed the current application and influenced the future direction of the application.

Chapter 7: Conclusions

As a conclusion of this report, we will list the limitations and challenges encountered during the development of the mental health application as well as any potential plans for the future.

7.1 Key Achievements

Defining success in any context is a complex concept, even more so in this particular project. Our definition of success in the context of this application would be how closely we matched the requirements of the Client, who introduced us to the topic with the following summary;

“MindfulStudent is a mobile app designed to support students' mental well-being through a holistic approach to mindfulness and stress relief. Key features include guided mindfulness sessions, interactive breathing exercises, a digital reflection journal, and gamified stress relief activities. The app offers tools for sleep improvement, progress tracking, and emergency mental health resources.

Additionally, MindfulStudent fosters community support through discussion forums and well-being challenges. It provides tailored mindfulness techniques for studying, encouraging positive habits and achievements. The app aims to promote mental well-being, offer practical stress relief tools, build a supportive community, and instil healthy habits for long-term mental health among students.”

As mentioned before, time and other resource constraints meant that we were not able to implement every mentioned aspect of the clients' wishes, but the primary goals of the system was:

- To be able to deploy a mobile application that is catered towards students hoping to manage their mental health.
- The application should offer holistic approaches towards reducing the stress levels of the target audience.
- To offer community support, to provide a safe environment for pupils who are facing challenges with respect to their mental health to be able to contact their peers with similar issues in order to form a sense of community.
- To provide a series of general tools that work in conjunction with each other to reduce the stress levels of students, rather than focusing on primarily a set action.

Based upon these goals, and in collaboration with our client, we were able to create a comprehensive list of requirements for the deliverable, and were able to rank these requirements based upon their importance, with requirements labelled as can be seen in Section 2.4. We can claim that we have fulfilled all the must and should have requirements requested by our client,

and a significant amount of the could-haves too. We have created an outline towards achieving mentioned requirements that we were unable to complete in the given timeframe.

We are proud to state that we believe each of these key goals has been implemented to a reasonable degree. We offer a deployable mobile application developed using Flutter for mobile applications, offer services like Breathing Exercises, Journaling, Task Tracking & Sleep Tracking which in conjunction with each other aim to enhance the current levels of mental health of students, and we offer a great Chat feature that allows you to find and connect with other users that may be suffering from similar mental issues, as well as contact mental health professionals should you need to. We believe that the general sense of community we wanted to achieve has been implemented, with a great deal of effort taking place to ensure items like profile images and reactions to messages were possible in order to generate a “social” effect on the user experience.

Another key concept that we focused on is that we have left the project open to future improvements. It goes without saying that gathering requirements, developing the product, ensuring development meets client expectations, testing and documenting an entire application within 10 weeks is a difficult task, yet leaving the project open to future improvements by using generic coding techniques has meant that if the project needs to be deployed in the future, it can be relatively easy to make changes to its current form to meet changing requirements. For example, changing servers from our current hosting provider to a self-hosted server would simply require changing the target URL on one line of code. Integration of other services apart from Fitbit for sleep tracking is also relatively easy since the code used in such areas is kept as versatile as possible. The key achievement in this context is that the deliverable has not reached its peak, but rather can be modified to change demand to meet new expectations.

Another key area that we are especially proud of is the general inclusion of stakeholders in the project development process. We were in regular contact with our client, who was regularly notified of the stage of development we were in, what the current application looks like and if any changes need to be made. We were also gathering input from another group developing a similar project with the client, as well as bi-weekly meetings with a larger collection of groups to explain our progress, changing goals and asking for feedback. We extended this by carrying out a user survey where we gathered information about users in order to better understand what their expectations from a mobile application were. We expanded on this by carrying out usability tests where users were given an early version of the application and asked to provide feedback based upon their usage, and the feedback they provided noticeably changed the final deliverable, as can be seen in the Usability Testing section of the report. The primary achievement in this respect is that the deliverable is not merely a representation of our version of what the application should be like, but rather a version created in collaboration with several parties and considering various perspectives.

7.2 Limitations of the System

While developing our application, we encountered difficulties along the way and a small fraction of them could not be resolved in the scope of our project. In this section we will briefly cover some of the difficulties we plan to deal with on a later stage of our product's development. Firstly, the technology stack we used, Supabase, was crucial for meeting our goals and is responsible for the healthy behaviour of the application. Following the current trajectory, however, in case MindfulStudent gains traction the pricing model we are using will be quickly blown out of proportion, leaving the responsible party with a huge bill to pay. Having in mind that all legal rights and ownerships are submitted to the University of Twente. Considering the hosting power of the legal entity such an expense would be absurd, however due to a variety of different constraints we could not prepare a model that works in favour of the university in the scope of the project. That said, Supabase has amazing documentation on how to self-host their entire infrastructure, and we took this into consideration by making the back-end server used by the application easy to adjust.

Additionally, as one of our initial requirements we had to introduce an animated character to assist with the breathing exercises, and potentially show up periodically and represent our application. Using a copyrighted character was out of the question for obvious reasons, therefore we had to consider either using characters in the public domain or creating a character of our own. The current laws state that characters are under copyright for 75 years, therefore in case we decided to use existing art we would have to go with characters from 1951. That seemed quite outdated and not at all in line with our goals, so that option was quickly dissolved. Due to developments of key features and optimising functionality we were unable to introduce any character made by our own team. In case this remains to be among the requirements, commissioning an artist to provide custom characters might want to be considered.

In its current state, the application has a few minor bugs that users may encounter, which does not necessarily prevent usage of the app but may show up to cause minor nuisance towards users. These are mentioned below;

- Some users noticed the application crashing if the user attempts to log in to the system with a FitBit account with no sleep data available. This is a relatively minor issue and could be resolved rather simply, but unfortunately was discovered too late to address in the current deliverable.
- Navigation is a bit inconsistent sometimes, does not make sense to all users. ex: sometimes when opening another screen back brings you to home, other times it closes the app.
- In the chat screen, if you have multiple incoming or outgoing connection requests and you interact with any of the requests that is not the most bottom one, the request below it

will take the place of the one that was interacted with but its details will not change, making it look like the one below it was accepted or denied.

7.3 Future Scope of the Project

Due to requirements and time constraints MindfulStudent is currently available only for Android users. Our team agrees that it is a crucial step to release our application for iOS as well, in order to reach a broader audience that would benefit from the project.

Additionally, as stated above, ownership will be handed to UTwente, where the university should be capable of handling the traffic of the app and host it on its own servers. While right now the hosting server we are utilising is part of the free package offered by Supabase, it can quite easily be switched towards a self-hosted server with larger capacity to suit a larger application.

Another important step would be to add frequent updates, releasing new content, in order to keep our users engaged in using MindfulStudent in the long run. Currently, ideas for such new features fall in the lines of gamification elements that come in the form of an in-game currency that could be used to purchase exclusive customizable items, such as different UI colour palettes, titles, banners and more.

On another note, as stated above, it was challenging to find a character in the public domain to use for our application, so a necessary step for future development, as long as we stick with that direction, would be to create digital art for our application in the form of different avatars, templates, etc. and add them to the in-game shop in future updates.

Last but not least, we hope to accompany each and every one of our clients on their spiritual and mental journeys, by adding new relaxing features such as guided book reading sessions and book recommendations, a symptom checker and psychoeducational resources.

To summarise the key areas of improvement for the app looking into the future:

- Switch from the server provided by Supabase as part of the free plan to a more specialised self hosted server that can handle larger amounts of data, while still using Supabase for the backend. This can be done relatively easily by referring to the URL of the new server in the code, a change of just one line.
- Currently, the application has not been deployed to the Play Store. Applications uploaded to Google Play are rigorously tested and due to the time available to us, we were unable to deploy the application before the end of the project. For ease of installation, it might be desirable to submit the application to Google so that users can install it on their devices using the regular method of installing new apps.

- Add book suggestions and tracking for users. This functionality would allow users to add books that they have read/are reading into the application. The app would then extract the genre of books the user is interested in, and generate recommendations based upon that. Furthermore, we could use this feature in conjunction with the “currency” element of gamification mentioned to further encourage users to read more, as this has shown to improve the mental health of students. Rather than creating a complex model to extract and evaluate genres ourselves, we were looking to integrate AI services offered by Open AI to request a list of recommendations based upon the user entry, which is a much simpler approach.
- Guided Study Sessions would be a very useful introduction to the application. Often students may exhaust themselves by studying long sessions in periods of high stress, leading to poor mental and physical health. There exist several new solutions to this, namely study sessions like “Pomodoro” or “50-10”, which suggests breaks at regular intervals to avoid tiring oneself. We may implement functionality that offers notifications to suggest these sessions based upon student preference. This can be done relatively simply by using the template of the Breathing Exercises screen with some modifications.
- Currently we are using Fitbit smartwatches for sleep tracking, but not all users in the user survey were comfortable with wearing a smartwatch to sleep. In the future, it would be beneficial to have integration with other services such as Google Fit or Apple Health to provide users with greater flexibility in terms of their sleep tracking. As mentioned before, the code is kept as generic as possible to allow for these services to be integrated later.
- While we do have the emails of users right now, we do not authenticate these or offer account recovery options in case the user forgot their credentials. These areas need to be addressed and possibly allow for sign up with mainstream services like Google/Microsoft, as recommended by our usability testing participants.
- Gamification is a great element that may be added to motivate users to take part in more exercises to improve their mental well being. The planned strategy would be to implement a digital currency, with additional points being offered for taking part in the offered services. The currency may be used to purchase elements like reactions to messages, new themes, avatars etc.
- Perform widget testing to ensure that clicking on each widget performs the function that we expect it to.

References

Broglia, E., Millings, A., & Barckham, M. (2017, August 1). Challenges to addressing student mental health in embedded counselling services: a survey of UK higher and further education institutions. *BRITISH JOURNAL OF GUIDANCE & COUNSELLING*, 46(4), 441-455.

<https://www.tandfonline.com/doi/epdf/10.1080/03069885.2017.1370695?src=getfr>

Zivin, K., Eisenburg, D., Gollust, S. E., & Golberstein, E. (2009, October). Persistence of mental health problems and needs in a college student population. *Journal of Effective Disorders*, 117(3), 180-185.

https://www.sciencedirect.com/science/article/pii/S0165032709000093?casa_token=U-UdkQtHfIAAAAAA:SUtC6LP-W0uf8ph7cqYqyOf8KYzsfTgUU_ownp0CJ1soz_A2rkTpl8PAcNy7FGT0bifedaY3hw

McLachlan, F., & Leng, X. (2020). Colour here, there, and in-between—Placemaking and wayfinding in mental health environments. *Color Research & Application*, 46(1), 125-139.

<https://doi.org/10.1002/col.22570>

Glossary

This section provides a glossary of commonly used terms. It can be used as a reference where the meaning of a term is not directly explained in the text.

Bcrypt: Bcrypt is a password hashing function. It is designed to incorporate a salt and has a variable number of iterations; this allows administrators to gradually increase the iterations in order to make it more difficult to crack.

CLS: In the context of PostgreSQL, Column-Level Security is an authorization policy that grants or denies users access to a specific column in a database table. It does not support custom SQL conditions.

DBMS: DataBase Management System. Examples of this include MySQL, PostgreSQL and Microsoft SQL Server.

Deno: Similar to NodeJS, Deno is a JavaScript runtime but optimised for security. It is designed to run on edge systems and has a fast startup time. It is also able to directly execute TypeScript code without transpiling it to JavaScript first. <https://deno.com/>

GoTrue: GoTrue is an authentication module that integrates with PostgreSQL and PostgREST. <https://github.com/supabase/auth>

HIBP: “Have I Been Pwned” is a service that scrapes the web for leaked passwords. It provides an API that allows other services to check whether emails or password hashes have been compromised in historical data breaches. <https://haveibeenpwned.com/>

HIPAA: HIPAA is US legislation mainly used in the health industry that mainly aims to protect users’ personal data when it comes to medical information. <https://www.hhs.gov/hipaa/index.html>

JWT: JSON Web Tokens consist of three base64-encoded objects that can be used for authorization purposes. They are signed using a private key on the server and can be verified by anyone given the correct public key. This makes them suitable for edge deployments, as there is no central database needed to verify the user making the request. <https://jwt.io/>

OAuth2: An Open Authorization standard. <https://oauth.net/2/>

pg_cron: pg_cron is a PostgreSQL extension that allows execution of certain PostgreSQL queries according to some predefined schedule. https://github.com/citusdata/pg_cron

PostgreSQL / Postgres: A wildly popular, open source Database Management System. <https://www.postgresql.org/>

PostgreSQL trigger: In the context of PostgreSQL, a trigger is an instruction for the DBMS to execute a predefined function under a certain condition. For example, it could run a function that executes a specific task whenever an INSERT occurs on a specific table.

<https://www.postgresql.org/docs/current/sql-createtrigger.html>

PostgREST: PostgREST is a layer around PostgreSQL, allowing the execution of queries using simple HTTP REST requests. It also provides a WebSocket to allow clients to be notified about real-time modifications to the database. <https://postgrest.org/en/v12/>

Provider-Consumer: In the context of Flutter, the Provider-Consumer system is a framework that makes a clear separation between data providers and data consumers. It allows us to easily update our UI (consumer) if the underlying data changes.

<https://pub.dev/documentation/provider/latest/provider/Consumer-class.html>

RLS: In the context of PostgreSQL, Row-Level Security is an authorization policy that grants or denies users access to a specific row in a database table according to some SQL condition.

<https://www.postgresql.org/docs/current/sql-createpolicy.html>

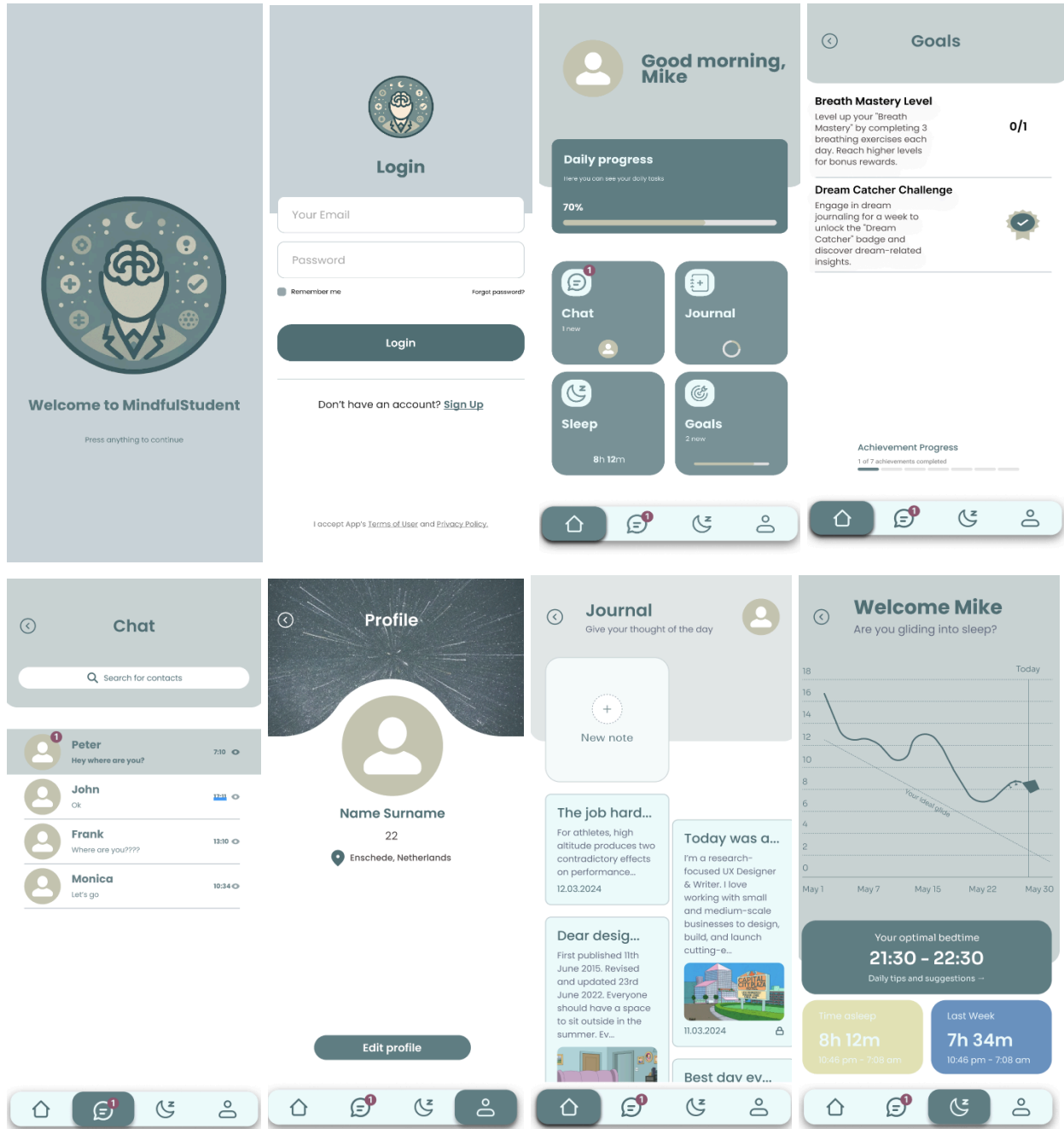
SOC 2: SOC2 is a certification granted by undergoing regular auditing procedures. It is mostly used by security-oriented businesses.

<https://www.imperva.com/learn/data-security/soc-2-compliance/>

TypeScript: TypeScript is a language developed by Microsoft. It is designed to be a shell around JavaScript by adding a static type checker. <https://www.typescriptlang.org/>

Appendix

A - Initial Design



Emergency contacts




Contact professionals

☎ **053 482 5510**



✉ secretariaatenschede@enschede.psyq.nl

📍 **PsyQ - Psychologist Enschede**
Hoedemakerplein 2, 7511 JP Enschede



Keep calm and stay in control

You can check your health with just one look.

Sign Up

Name

Your Email


Password

Confirm Password

Create account

Already have an account? [Login](#)

I accept App's [Terms of User](#) and [Privacy Policy](#).



Your account has been created

Log in

Peter
last seen today at 7:20

March 3

Hey, what are you up to? 12:18pm ✓

Nm 12:24pm

Just lounging on your laptop 12:25pm

Mmm... why? 12:27pm ✓

Warming it up 12:28pm

That's not how viruses work 12:35pm ✓

No i'm good just asking 12:22am


Fair enough 12:40pm ✓

I locked myself out 2:01pm

You always do 2:02pm ✓

Message

Edit Profile



Name

Name Surname

Email

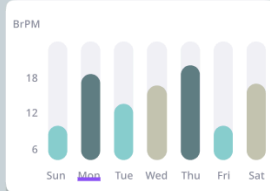
namesurname@gmail.com

Password

Save Changes

Breath Rate details

📅 04/03 - 11/03




Weekly summary

A person's respiratory rate is the number of breaths you take per minute. The normal respiration rate for an adult at rest is 12 to 20 breaths per minute. A respiration rate under 12 or over 25 breaths per minute while resting is considered abnormal.

Insights for you

1 Try to complete one Breath Training every day

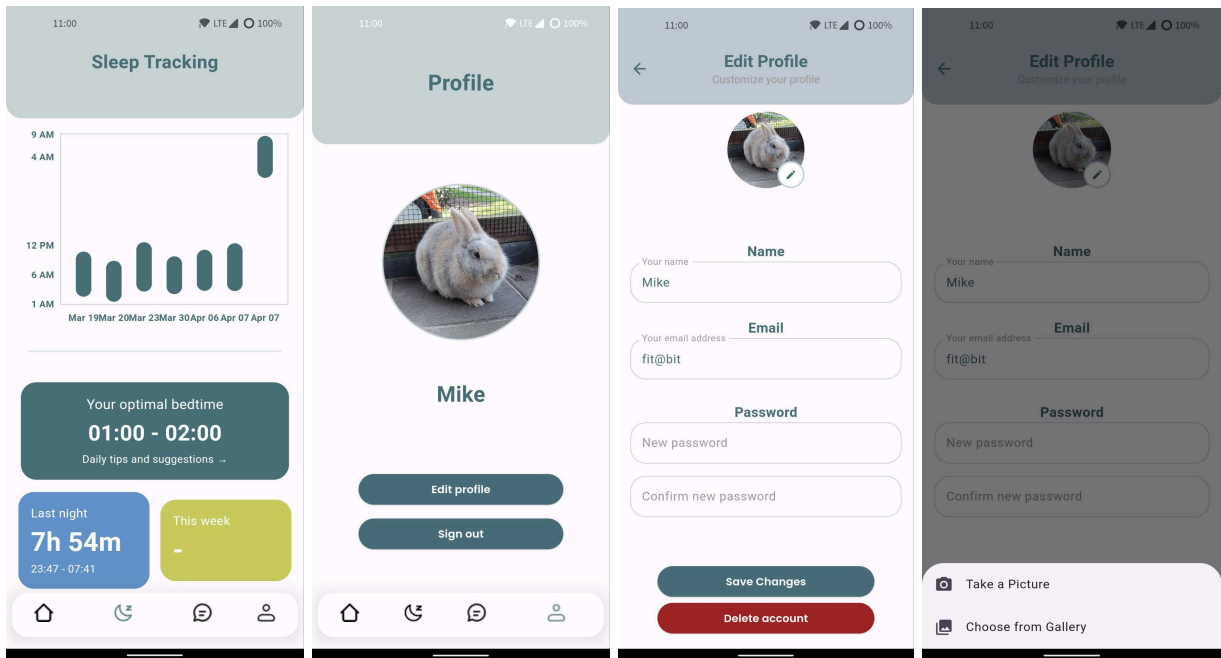
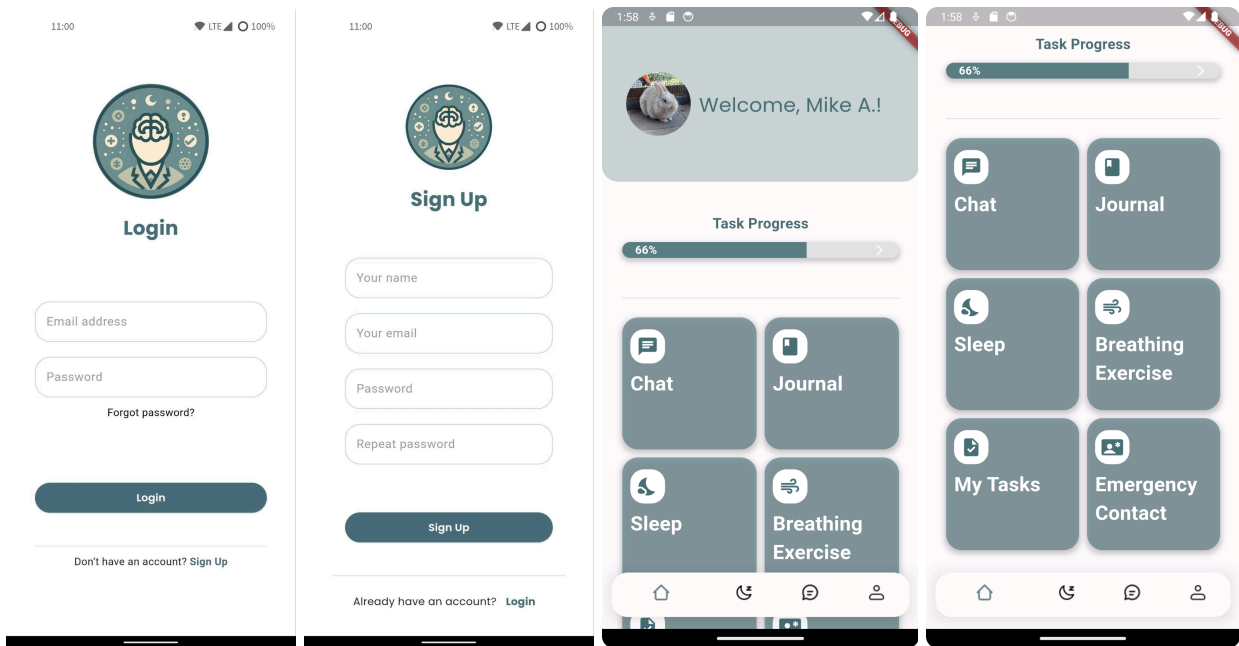


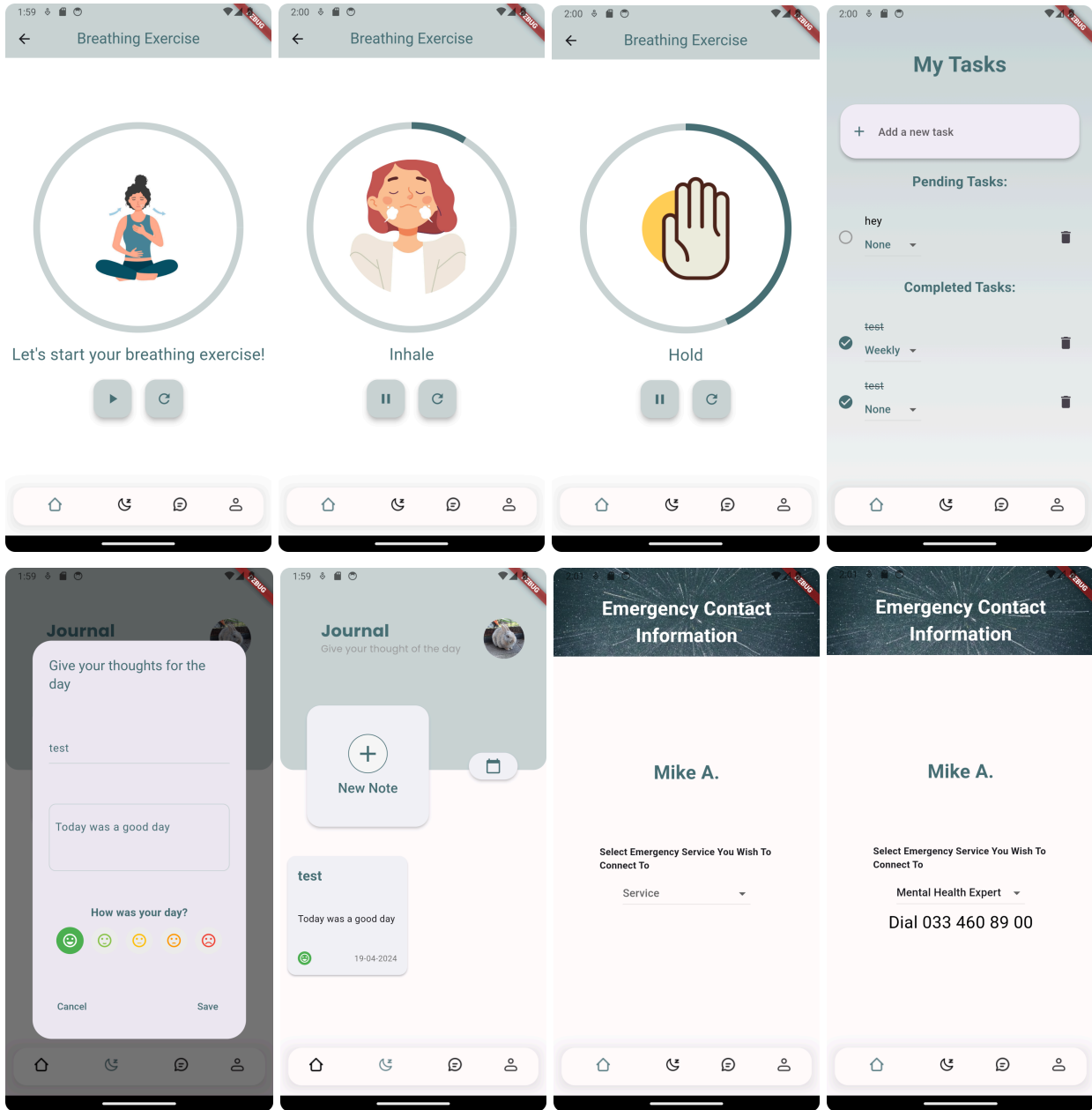
You have completed Breath Training

Time	5:00
Before	18 BrPM
After	8 BrPM

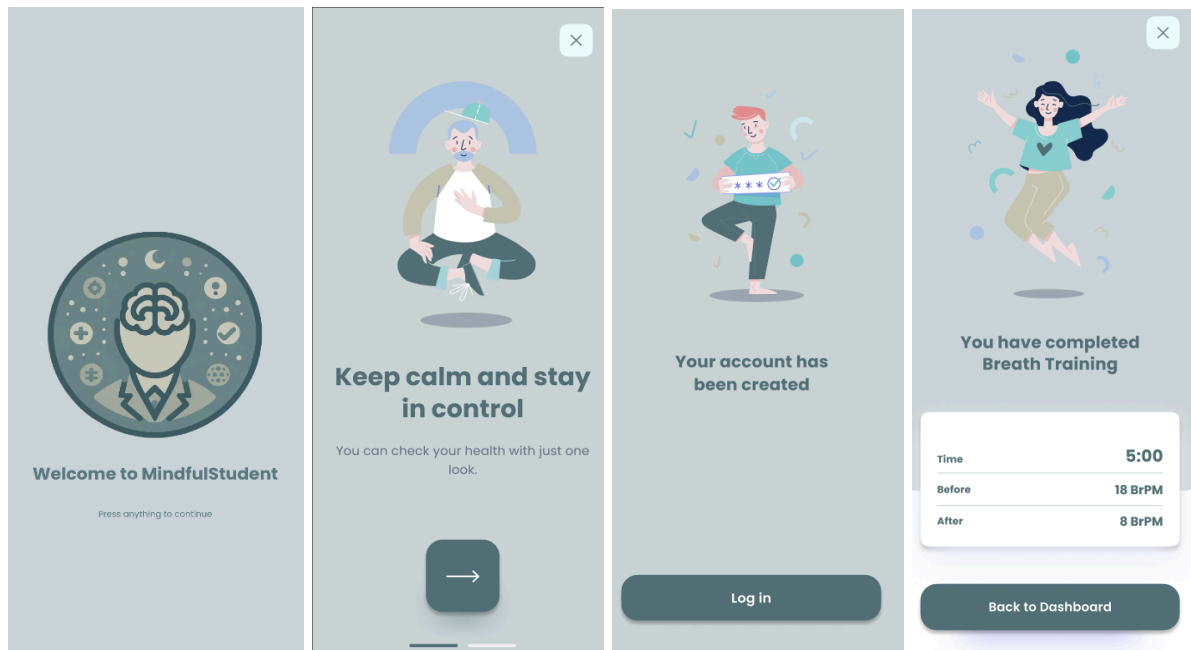
Back to Dashboard

B - Final Design





C - Discarded UI Pages



D - User Survey

As mentioned in the report, the primary purpose of the user survey was to gather key information in respect to the demographic that we serve. Within this document, we publish the method of the questionnaire we asked to fill out, how the participants were selected, the structure of the questionnaire, display the results and the interpretation of them.

Participants

The target group of the application is university going students. Therefore, we found that the most convenient method to send the questionnaire to the target group would be through asking our friends and colleagues to fill in the questionnaire. This would ensure that the results are only gathered from students who may potentially form part of our target group, while the fact that we personally knew the participants meant a more honest and sincere reply. As a future improvement however, we would like to be able to send the questionnaire to a wider audience in order to gain feedback from a bigger range of people.

Questionnaire

The questionnaire is broken down into several sections, with each section aiming to provide us with insight into a particular topic of interest. These sections, along with their purpose, are;

Demographic Data

Get to know the person filling in the form, their age, course, year of study etc. This would allow us to be able to identify subgroups within the target population

Sleep Queries

Get insight into the sleeping aspect of student life. Items like identifying if subjects find it difficult to sleep, spread of insomnia amongst students etc.

General Mental Health Questions

Gather details about some rather personal details about the student's mental health, like if they are speaking to an expert. In order to make the subject feel as comfortable as possible, we do not require them to fill in their name or email address to make all responses completely anonymous.

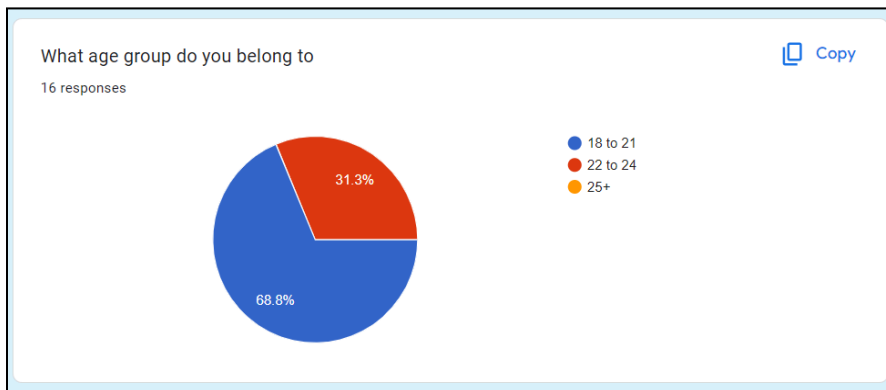
Application Queries

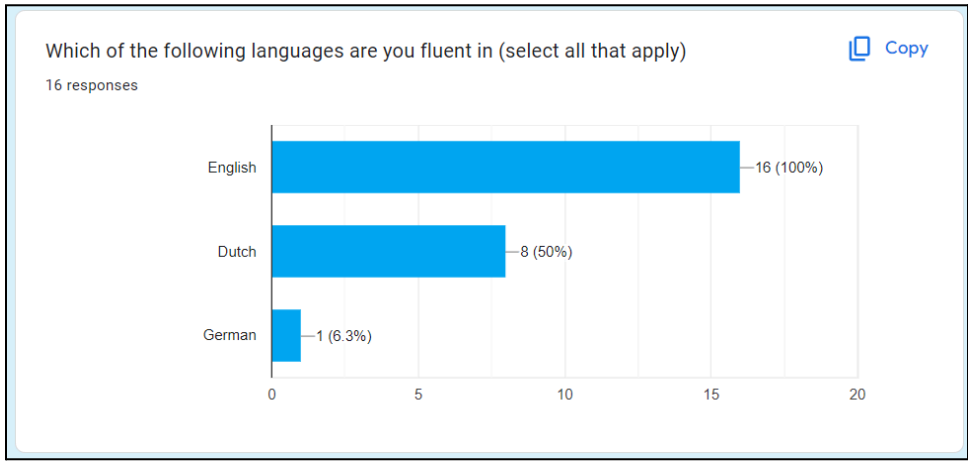
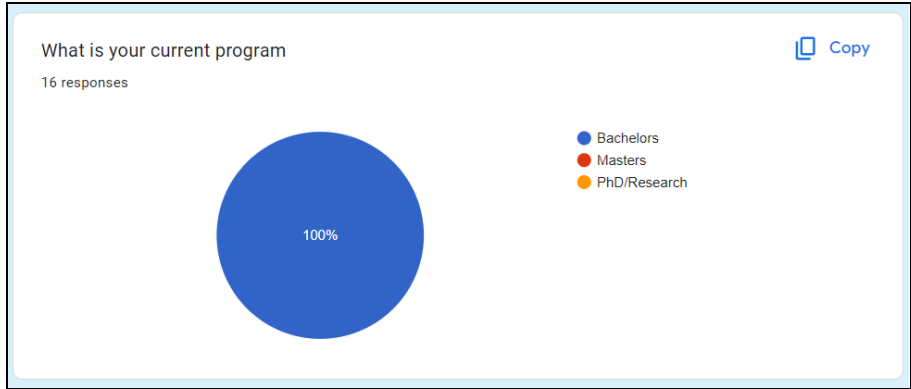
After all, the application will need to be installed by the target audience. This section involved asking questions about the general type of application features the audience prefers, like what colour scheme they appreciate or what costs they'd be willing to pay.

Results

The results for each of the sections are as follows;

Demographic Data:





List any other language that you are fluent in

7 responses

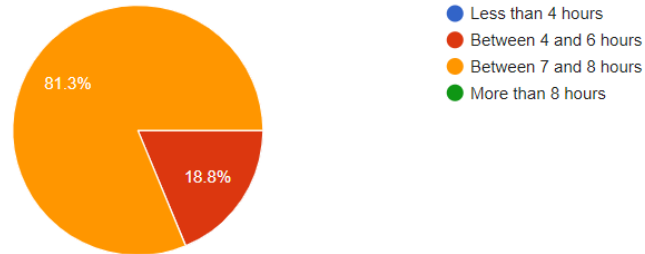
- Urdu
- Arabic
- Hindi, Icelandic, Bengali
- Spanish
- Bulgarian
- Japanese

Sleep Queries:

How many hours of sleep in a day would you say you receive on average?

 Copy

16 responses

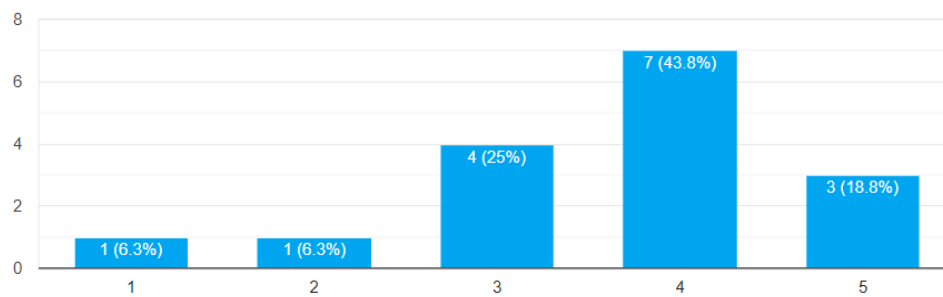


For the following section, please select a number ranging from 1 to 5, with 1 being you do not agree with the statement at all and 5 being you completely agree with the statement.

"I feel that I get a sufficient amount of sleep every night"

 Copy

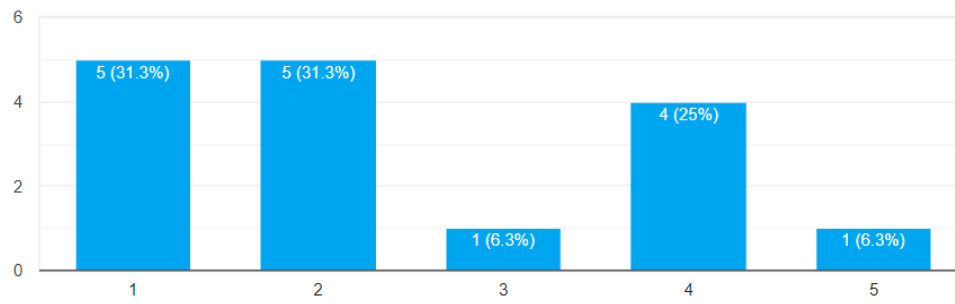
16 responses



"I find it difficult to fall asleep during most nights"

 Copy

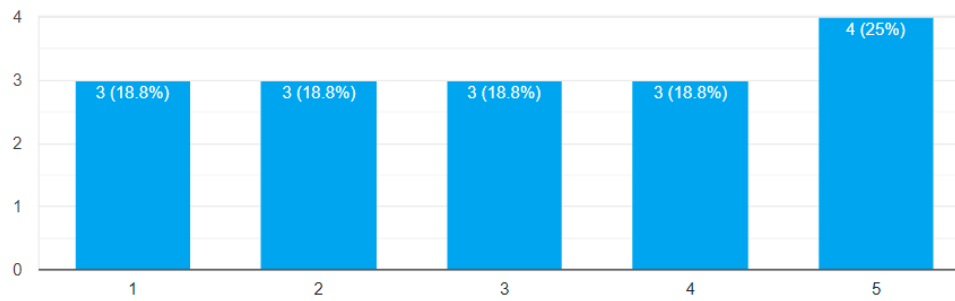
16 responses

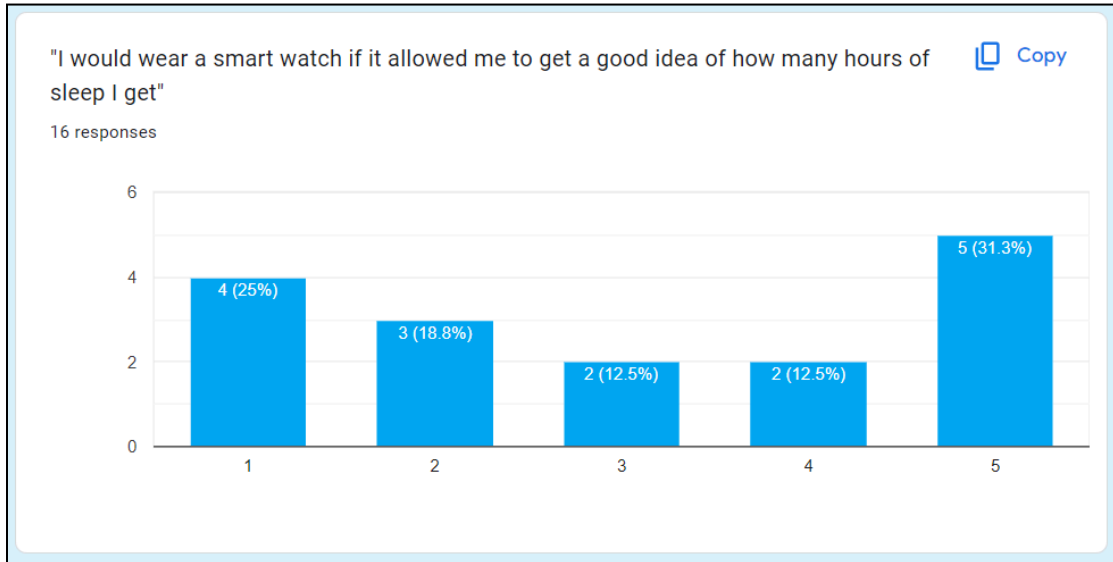
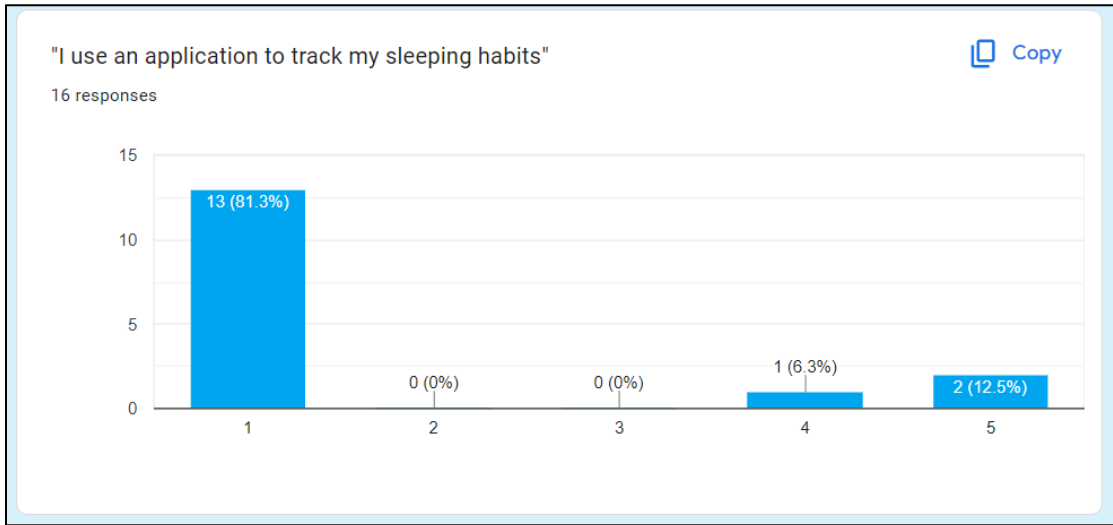
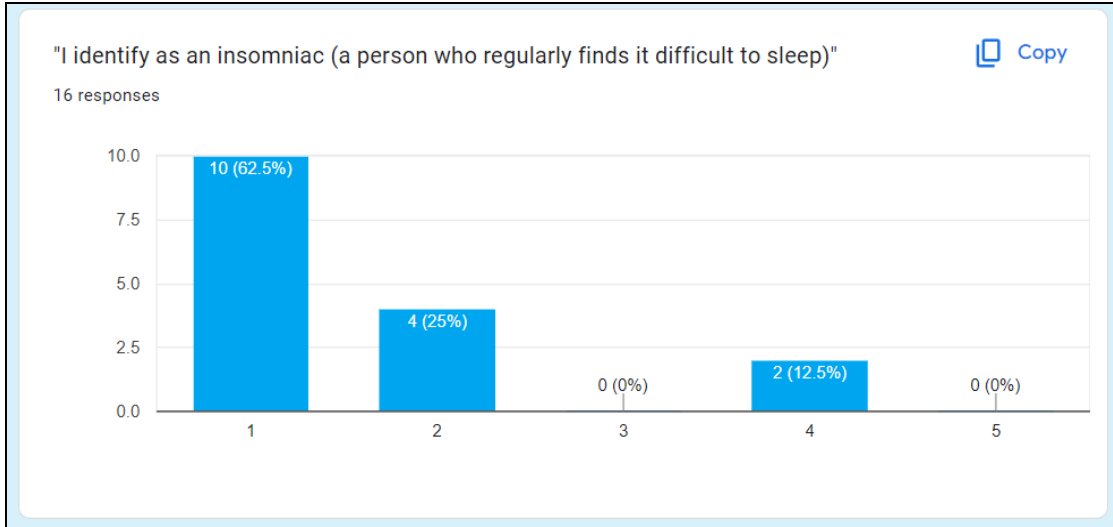


"I feel it is more difficult to sleep during exam periods/near assignment deadlines"

 Copy

16 responses

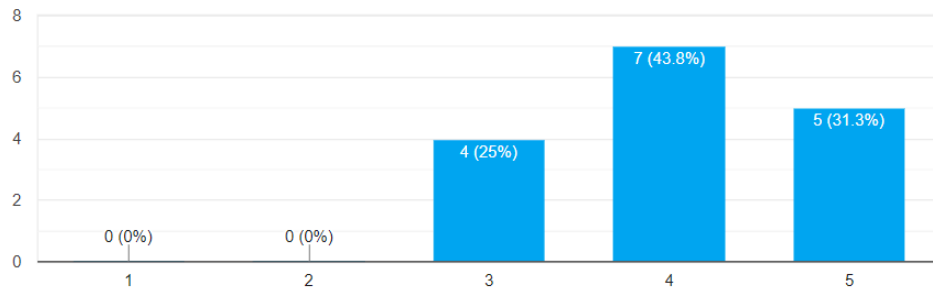




"The hours of sleep I get affects my mood and motivation for the rest of my day"

 Copy

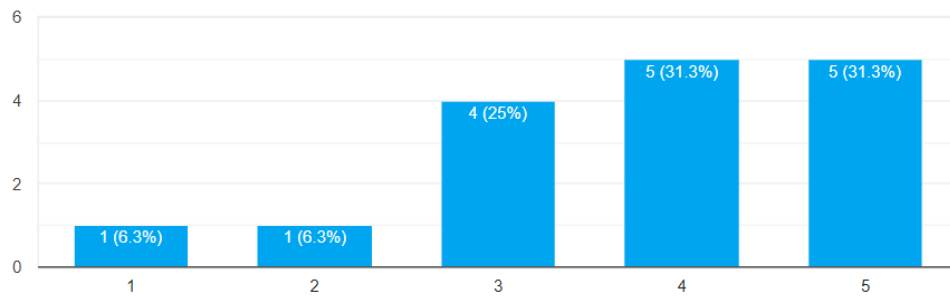
16 responses



"It would be useful to know how many hours of sleep I got on a certain night"

 Copy

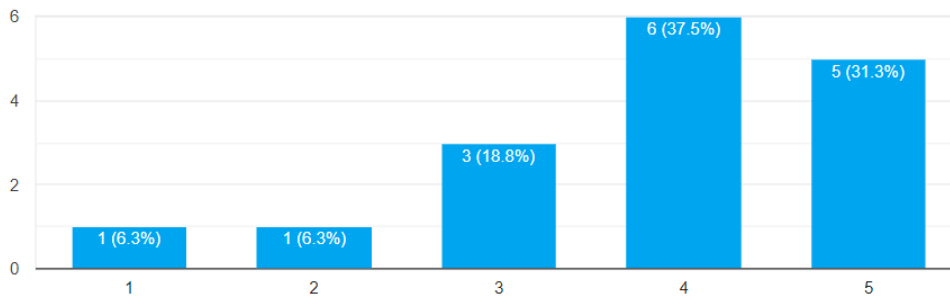
16 responses



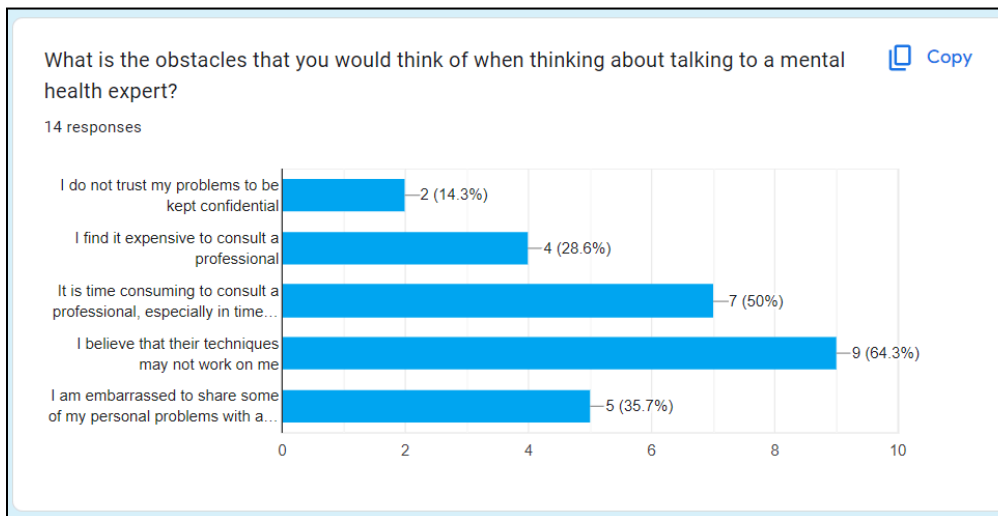
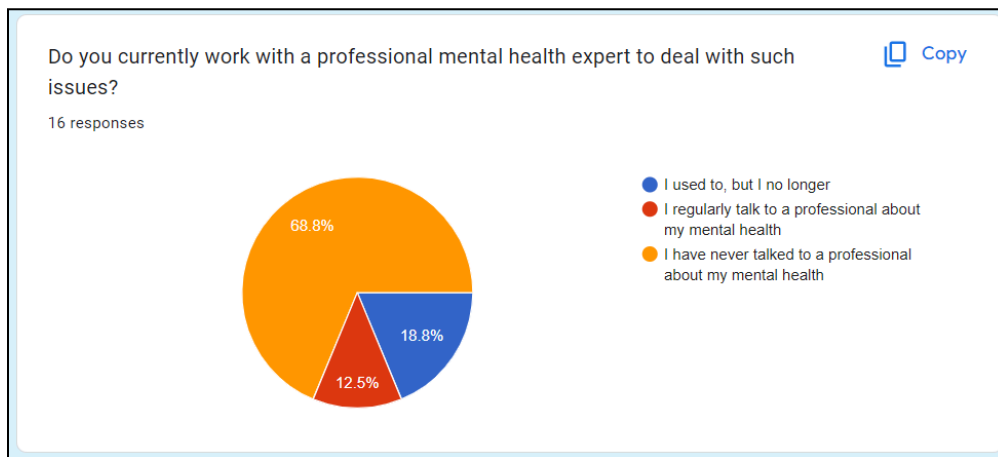
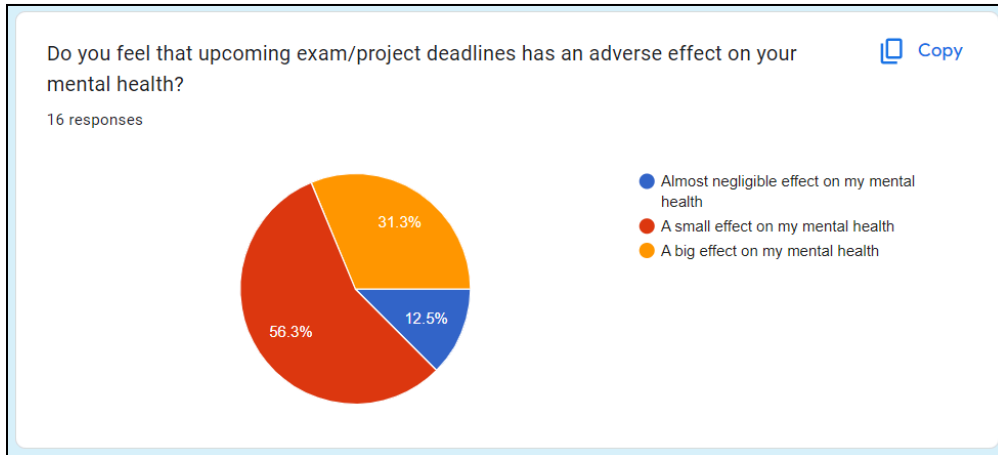
"It would be useful to see how my sleeping trends have progressed over a period of weeks"

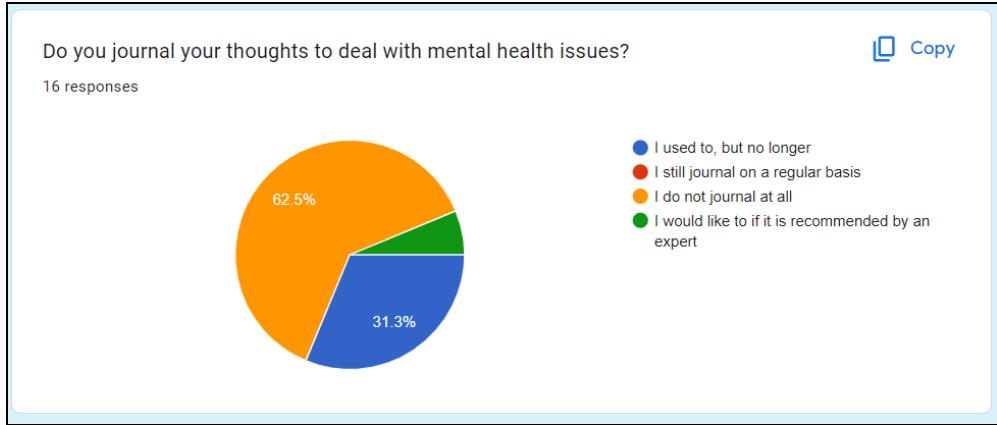
 Copy

16 responses

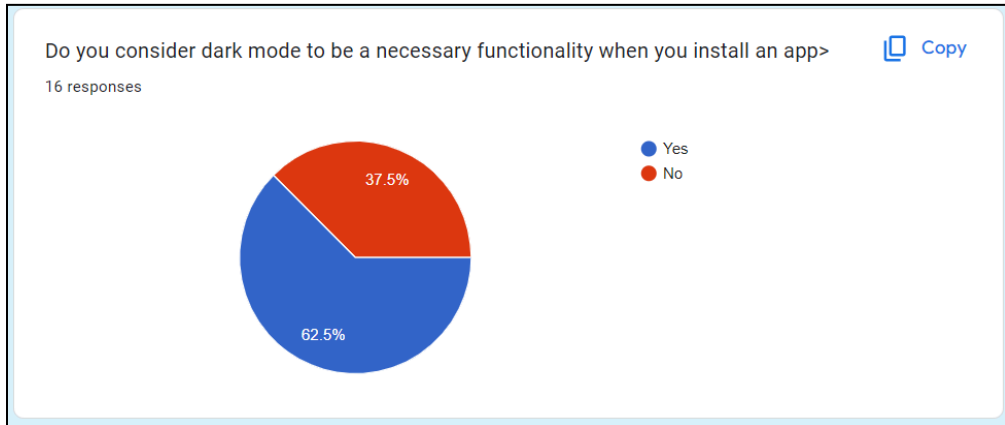
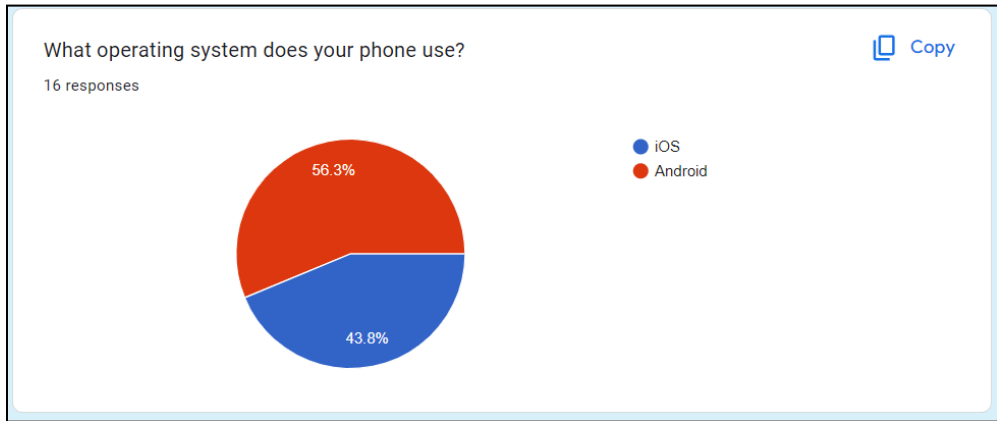


General Mental Health Questions





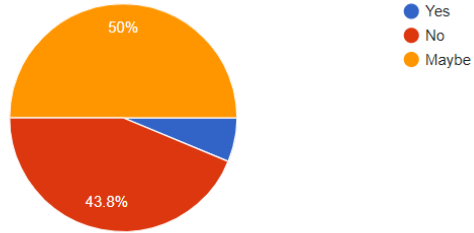
General Application Questions:



Would you consider paying a small amount for an application that offers you with help to deal with mental health?

 Copy

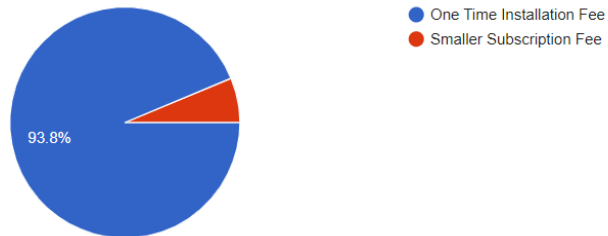
16 responses



If you have to pay for an application as such, would you prefer a one time installation fee, or a smaller subscription fee.

 Copy

16 responses



Which of the following colors do you believe appear to stimulate a "relaxing atmosphere"?

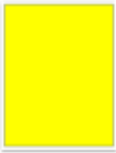
Option 1



Option 2



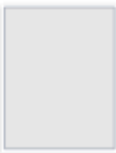
Option 3



Option 4

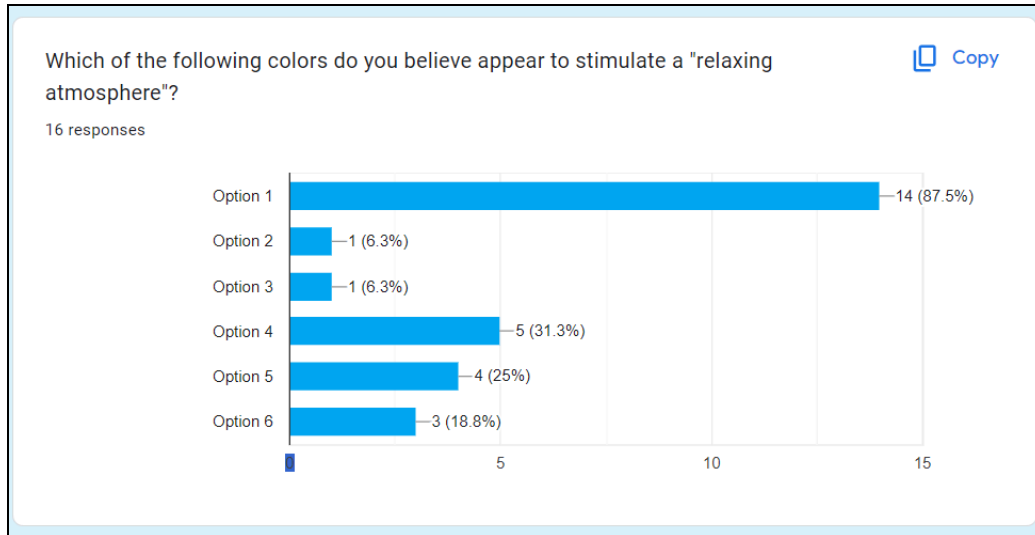


Option 5



Option 6





Interpretations

The interpretations we took for each of the sections have been listed below.

Demographic Data

Most of the participants were within the age group that we anticipated would be the primary target group, ages 18-21. We believe that this is the age group that most new university going students would be likely to belong to, therefore be more likely to require help in terms of their mental health. While we do have some representation from ages 22 to 24, there is very little in terms of older ages, meaning that we may not have enough data to be able to cater specifically to the needs of PhD/Research students. As a future improvement, it would be nice to have a slightly larger representation group. This point is further emphasised with the current course program of participants, with all of them lying within the Bachelors ring, creating an element of bias. We discovered from this survey that all participants are fluent in English, which is currently the only language supported by the application. Yet so, roughly half the participants were fluent in Dutch, while other languages like German, Hindi, Spanish also represented which may be introduced in the future. The total number of participants for this particular study was 16. In order to increase the knowledge gained from the survey, as a potential future improvement we would perhaps like to collaborate with various universities within the Netherlands to gather data from Bachelors, Masters and Research students from a wide range of backgrounds.

Sleep Queries

We found out that a majority of participants were receiving between 7 and 8 hours of sleep, which is deemed healthy for people within the age group. No participant reported less than 4 hours of sleep. Furthermore, it was reported by more than 60% of the participants that they receive a sufficient amount of sleep every night, which may slightly reduce the impact of sleep tracking functionality in application, at least from what we initially anticipated. A similar amount

of people reported struggling with falling asleep and even fewer reported identifying as an insomniac. However, participants did show a noticeable change when asked if they find it difficult to sleep during periods of high stress, with that showing even distribution. Even more surprisingly, 80% of participants reported that they do not use any application to track their sleeping habits. All these responses meant that we must aim to incorporate some form of motivation for users to use the sleep tracking functionality that we offer, in the form of exploring the benefits of tracking sleep, showing changing patterns etc. This is greatly supported by the other questions, with almost 80% of respondents reporting it would be nice to know how many hours of sleep they received, 75% reporting the hours of sleep they receive impacts their mood for the day and 70% agreeing that it would be nice to know how their sleeping patterns progress over a period of time. Roughly 50% of participants reported a reluctance to wearing a smartwatch while sleeping to track their sleeping patterns, which is currently the only form of sleep tracking we offer. For future improvements, we may offer alternatives to this in order to gain support from this part of the population. The general takeaway from this section is that while sleep tracking is not very common amongst a lot of the users and they may not necessarily feel it is difficult to fall asleep, they would appreciate knowing their sleeping trends and it may be a factor that improves their mental health. The fact that some students report that their sleep is affected by periods of high stress and $\frac{3}{4}$ of the participants reporting that sleep does have an effect on their mental health, it can be fairly confidently be concluded that the sleep tracking feature will benefit the target population.

General Mental Health Questions

More than 85% of participants reported at least some level of impact approaching university related tasks had on their mental health, which highlights the need for applications in the nature of what we are currently developing. Alarmingly, almost 70% of participants reported to never speaking to a mental health expert in their entire lives, with only 12.5 % speaking to one on a regular basis. When asked why they may not approach an expert to speak in such regard, the primary reasons were that they believe their techniques may not necessarily improve their mental health and they do not have time to do so. The chat functionality to mental health experts greatly reduces the time taken to speak to an expert, addressing one of these concerns. In terms of improving the perception of speaking to mental health experts, we would wish to incorporate videos in the application encouraging users struggling with experts to speak to experts, as well as hiring high quality experts to ensure the ones that do approach feel fulfilled. Almost none of the participants reported to journaling on a regular basis, with even fewer reporting to do so if it is recommended by an expert. This suggests that journaling amongst students is not a high enough priority to be given too much time and resources in the development stage of the project.

Application Queries

At the current state, the application is only offered for Android users. While 56% of participants reported having an Android phone, there is still a significant population that goes unserved if the application is not available for IOS devices. This has meant that one of the key areas of

improvement for the application in the future is to make it available on the App Store. In collaboration with our client, and in keeping terms with the general theme of the project, we did not decide to offer a dark mode since we believe it may evoke negative connotations in the minds of the user. However, more than 60% of the respondents believe dark mode to be a necessary aspect of any mobile application, meaning it should be high on the list of future updates to the application. Furthermore, to hire experts/maintain servers etc we would require financial support, and selling user data like most the current state-of-the-art does does not seem ethical in context of the application therefore it would be preferable to ask clients to pay a small fee for using the application. However, almost 40% of the respondents reported that they would not pay for such an application. This would suggest a dilemma and we may require financial support from institutes to run the program, since very few respondents were actually certain to pay a fee to use the application. If we do require a fee, most participants would rather pay a larger one time fee than a regular, smaller subscription based service. Finally, in terms of what colour would evoke the most calming sensation within the user for the background, a light blue colour was selected, which will form the home screen for the application

E - Usability Study

Idea of the Study

The purpose of Usability Testing for the application was to gather insight about how the application would be perceived by members of our target audience, without the knowledge of the details of implementation for the application that the development team had. To accomplish this, we invited 3 participants from the target audience, with each of them studying at the University of Twente for 3 or more years, from the Bachelors program and within the age range from 21 to 22. Each of these participants was provided with a preliminary version of the application, and they were asked to use the various features of the application, just as the intended final user would. Based on the observations from how they use the app, coupled with them answering some of our questions about the application, we were able to gather some very useful information about the application. This included identifying some areas of confusion where the application may need to improve, getting feedback on the colour scheme and general theme of the application, gathering input for future improvements for the application and much more. The technique to conduct the interviews, the results and their interpretations form part of this document.

Technique

As mentioned previously, the subjects for the usability study were chosen very carefully. We had to ensure that they were enrolled in a university, that they could speak English fluently enough to understand the contents of the application and the questions. Based upon feedback from the project progress meeting with the other groups and module supervisors, we decided to ask 3 of our friends that are not part of the module to conduct these user tests. This was made to ensure that the participants would be willing to offer significant feedback from the interviews, and the fact that they are fellow Computer Science students that have seen usability tests conducted before may mean they are better suited towards answering these questions. Dates for the usability tests were decided, with one test on the morning of the 8th of April and two on the morning of the 9th. Rooms were booked to ensure that there would be minimal disturbances. A git pull was performed on the 6th and the features of that version were to be used for the tests. We decided to use an emulator for the tests rather than a phone since an Android Device was unavailable for the subject conducting the interview. Participants were thanked for their time and feedback, and provided snacks before the study started as a means of expressing). gratitude, and to ensure better feedback. At the start of the experiment, they were informed of the purpose of the study, how their feedback will improve the application and informed that they may leave the study at any time without having to give reason. After they have read and understood the conditions of the study, we ask for their consent to begin the interview. For each screen, we asked the users questions to rate usability on a scale of 1 to 5. At the end of each screen, we

asked the user if they had any suggested improvements for the usability and layout of that particular screen.

The Results

User Data

The first section dealt with personal data to allow us to better understand the type of subject. We will not be sharing the precise names, ages or nationalities of the subjects due to privacy concerns, but their age range, years spent in a university and daily usage of mobile phones devices ensures us that they are sufficiently suitable to form part of our target population.

Login Page

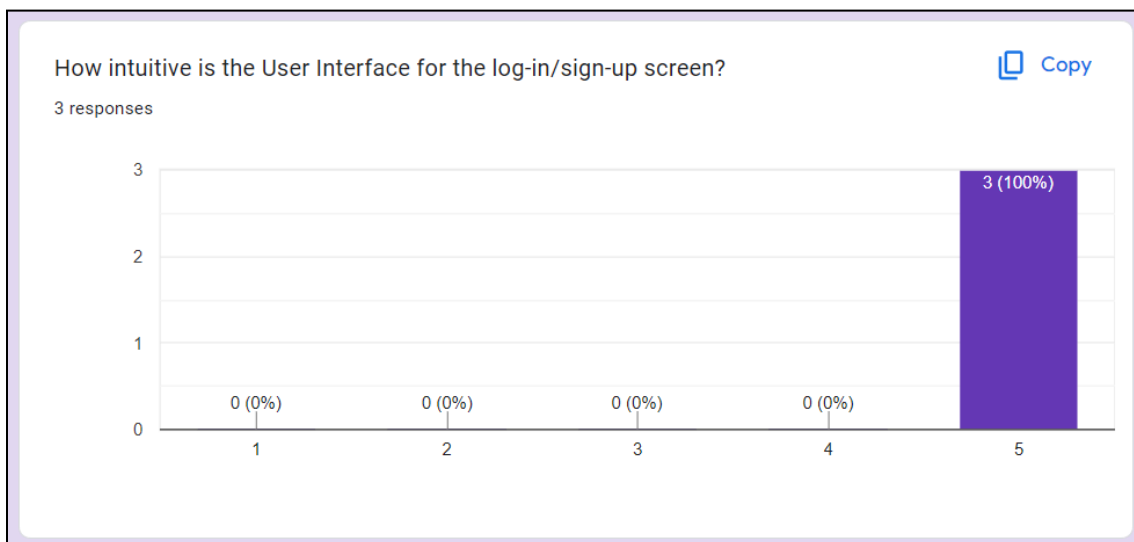
Users were asked to sign up to the application for the first time on the startup screen for the application.

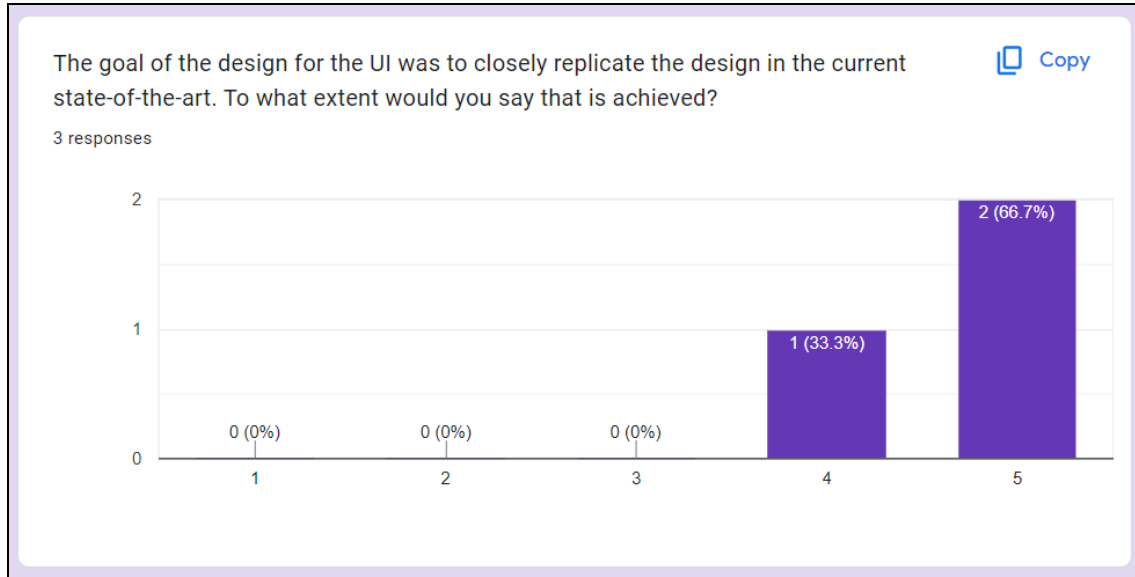
For creating the signup, we decided to inform them not to use any passwords they would use for their other accounts since although we are fairly confident in the security robustness of our application, we decided against potentially risking the experiment with leaked credentials.

After creating an account and logging in, the users were asked to rate the general intuitiveness of the application, giving a rating out of 5 for how easy and natural it was to create an account and login to the system.

After that, the users were asked how closely the signup system for the application matched the current state-of-the-art (meaning how similar did it feel to if you were to compare it to creating accounts with more commonly known services like Google).

Other feedback we received in this particular section was to include a sign in with Google/Microsoft option since some users feel that this is the easiest and most secure method of creating an account.





Home Page

One of the requirements of the application was to have a personalised greeting for when users enter the application, with the main goal being to uplift the mood of the individual. We asked the users if they noticed the personalised greeting that we implemented into the home page.

Another primary goal of the application was to find a colour scheme that would subconsciously reduce the anxiety levels of the users, by using light colours that would be associated with positive feelings. We asked the users if they noticed such an effect based on the theme of the home screen.

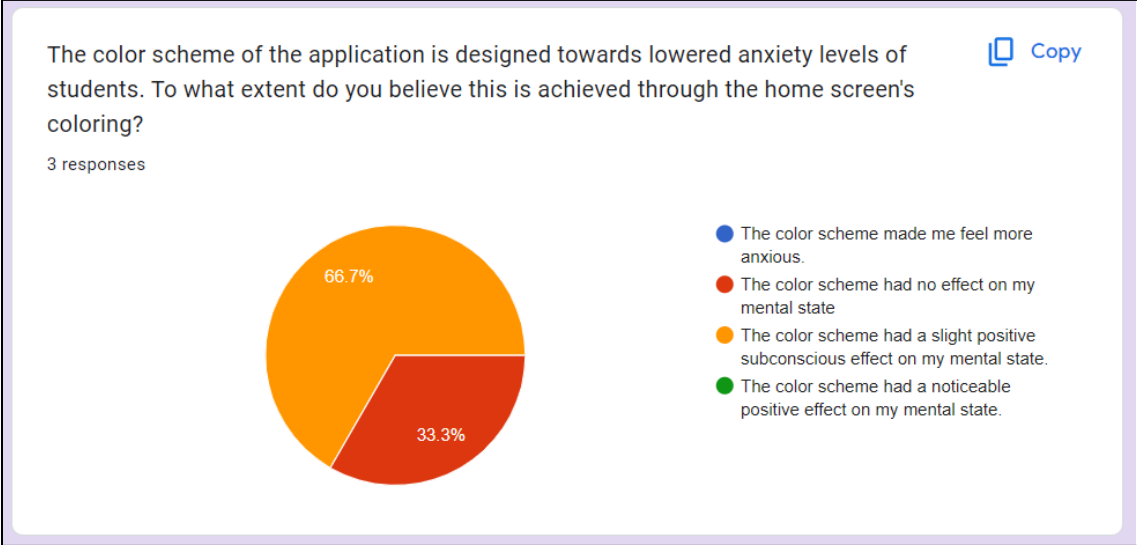
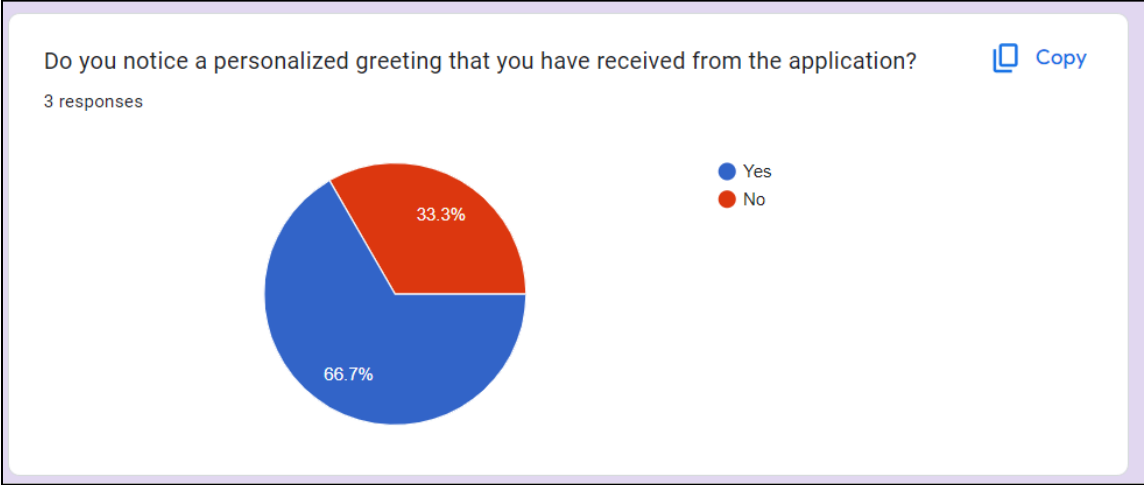
We offer several features on the home screen that are designed to improve the mental health of students. One of the main queries was how easy it is to navigate through the various features on offer on the home screen.

We asked to rate the general layout, structure and colour scheme of the application, in the sense that it looks professional, easy to navigate, easy to access etc.

Furthermore, a key design choice was to ensure that features that we believe the users would be most likely to use most frequently to be placed towards the top of the home screen, to increase accessibility. We asked the users, based on their preferences, to what extent this was achieved.

Lastly, the home screen has a bottom navigation bar, with a similar purpose of offering quick navigation to some of the most used features. We asked the users to provide their top 3 most important features from those on offer to place in the bottom navigation bar.

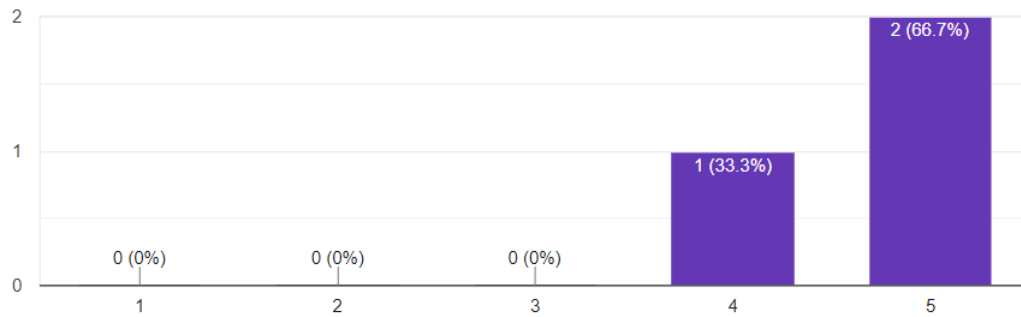
One user noticed that the icons for the home screen were repeated, which although small in size could lead to confusion for some users, as well as looking slightly unprofessional.



The application offers several features towards improving the mental health of the user. How easy is it to navigate through these features on the home screen?



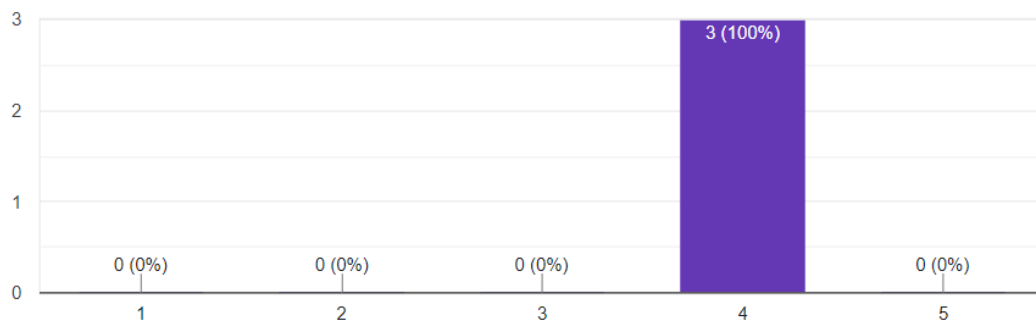
3 responses

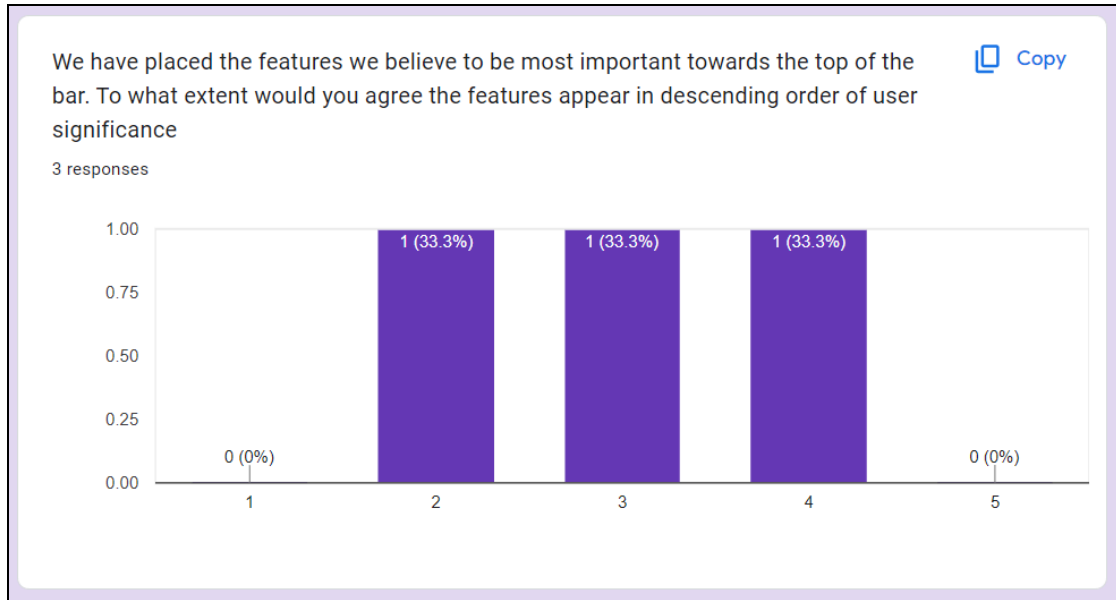


How would you rate the overall design of the home screen, in terms of coloring, access, layout etc.



3 responses





Can you list the 3 features from what we offer that you would find most important for a mental health app

3 responses

- Sleep, Goals, My tasks
- Journal, My task, Chat
- Chat, Journal and Goals

Journal

For the Journal page, we asked the users to create a journal entry with some sample data. We then asked them to rate how easy and intuitive it was to create the entry.

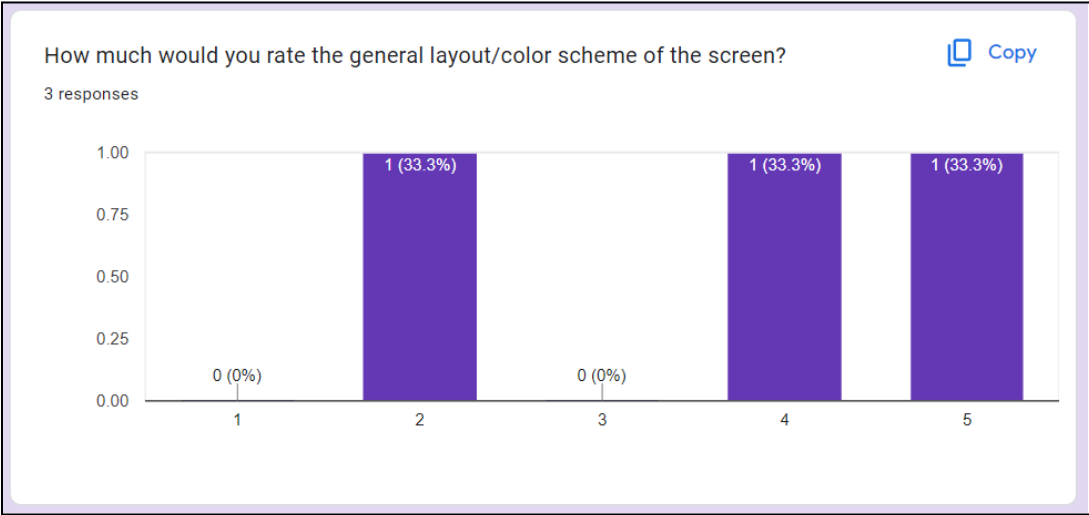
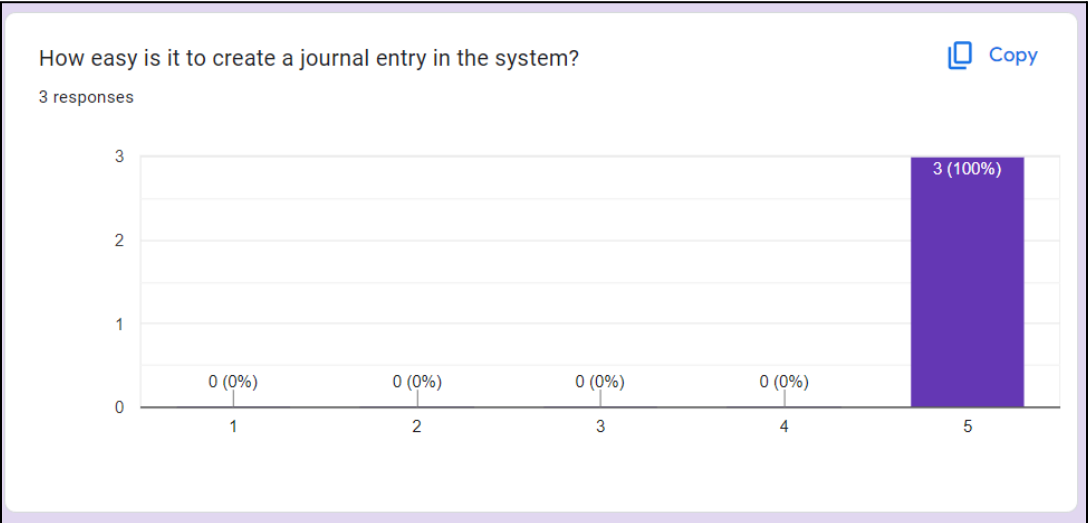
We then asked the user to rate the general layout and usability of this screen.

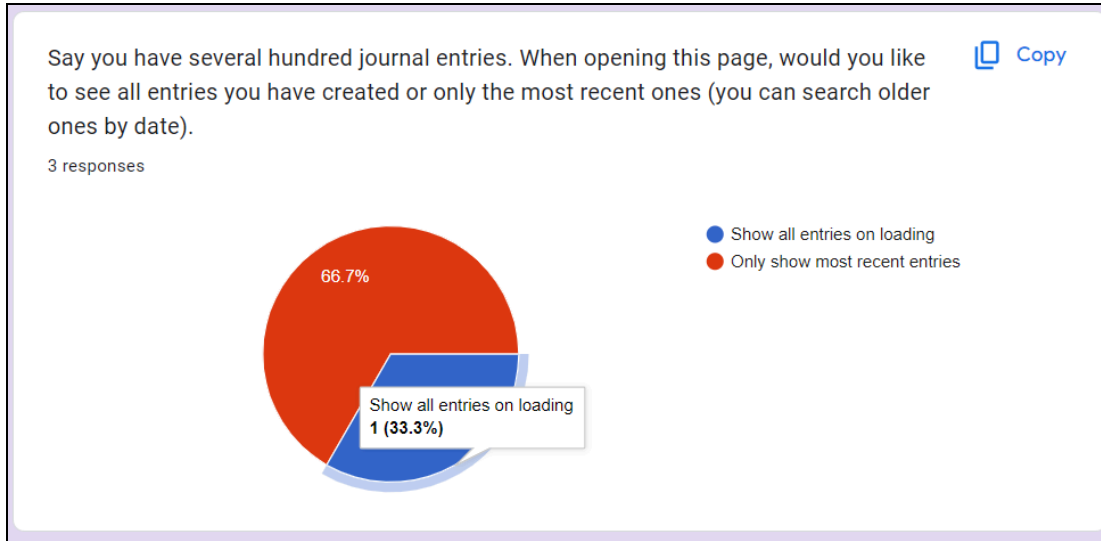
A key design question that we had was whether to display all journal entries upon loading of the screen, potentially overloading the user with hundreds of entries if they journal regularly, or only to load the most recent ones with the option to filter by date.

One of the participants mentioned that the Title section seemed confusing and redundant, and that they'd rather not have to think of creating a title for the entry.

Furthermore, a participant mentioned that a possible improvement would be to have a colour indicator for each entry to rate how their day was going, like a traffic light system. This would be a good way of scrolling through entries to see how their mood was at a given period of time.

Users noticed the lack of option to delete journal entries, which they found to be a useful feature that should be included.





Breathing Exercises

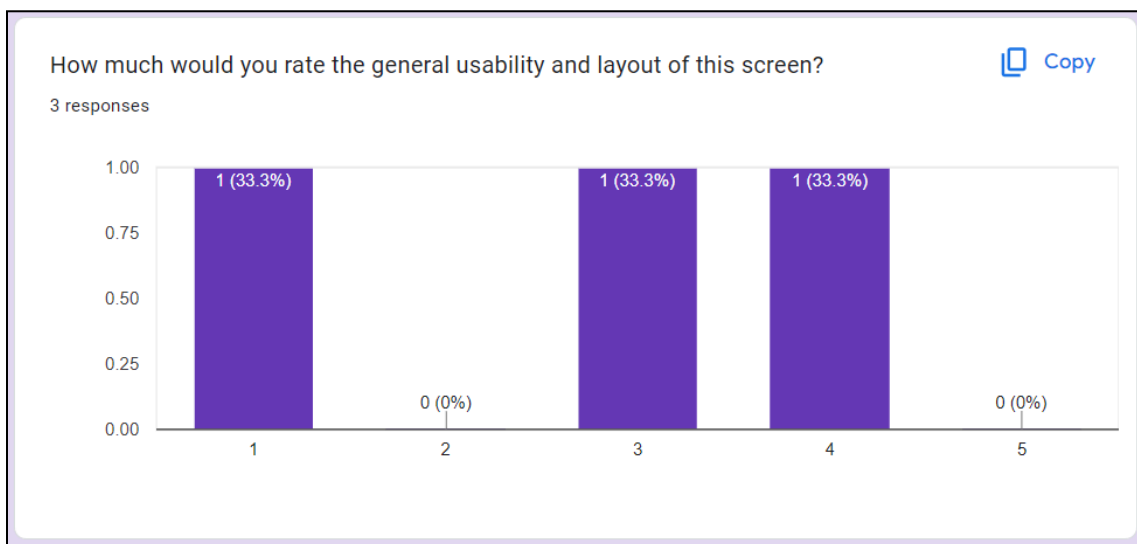
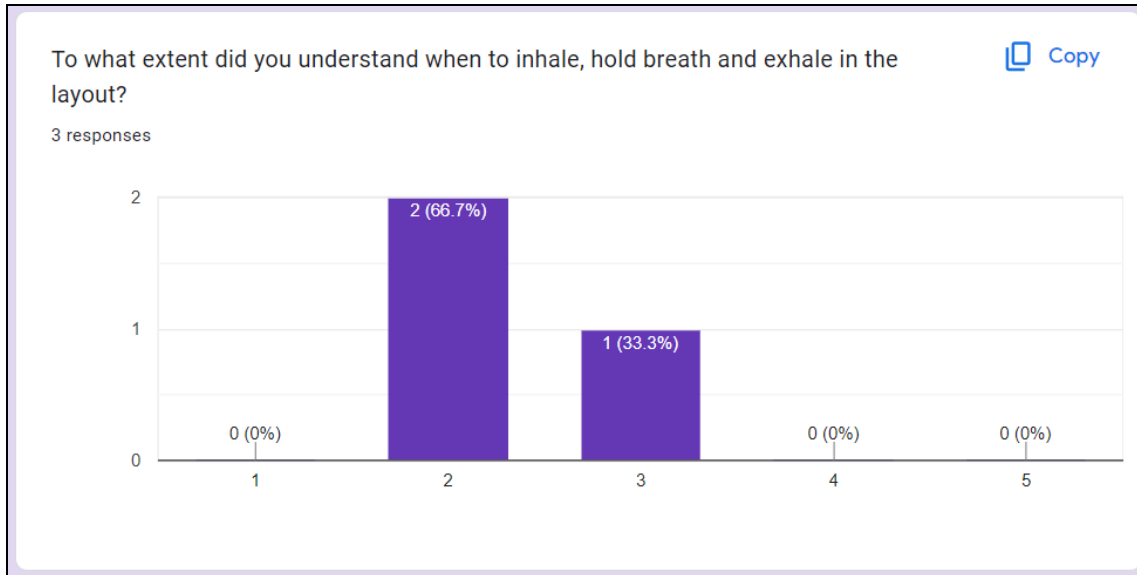
The next item to test was the Breathing Exercises functionality. Users were simply asked to start a breathing exercise, and we evaluated their interaction with the screen and gathered feedback about its general usability, layout and the effectiveness of the exercise.

The general rating of knowing how to start, pause and restart the exercise were rather poor.

The rating of the general layout of the screen was also rather poor.

All participants were confused since clicking on the “Start” at the centre circle did not indeed start the exercise.

Furthermore, most of the participants were unaware of how long to inhale, and some reported difficulty in taking part of the exercise overall due to the confusion.

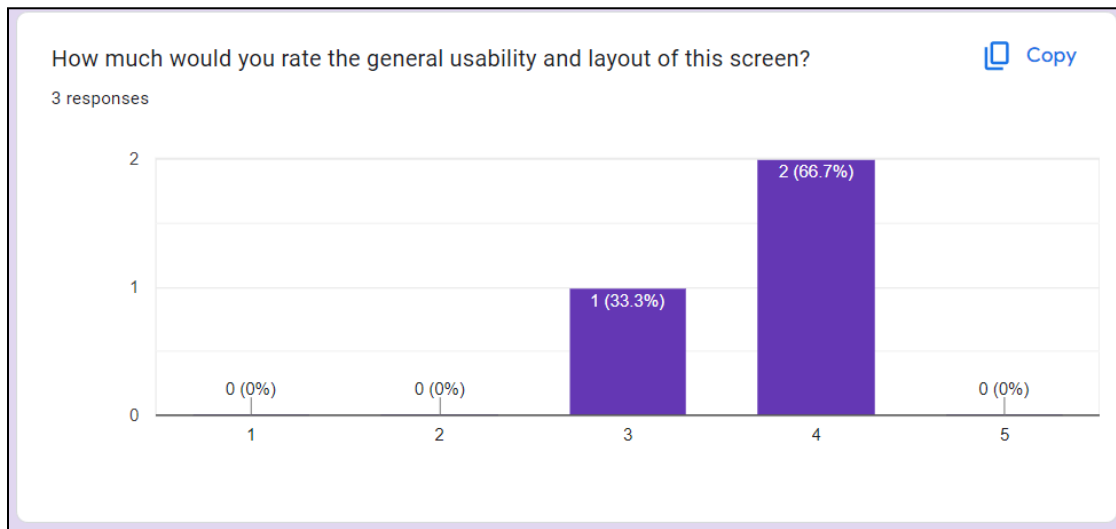
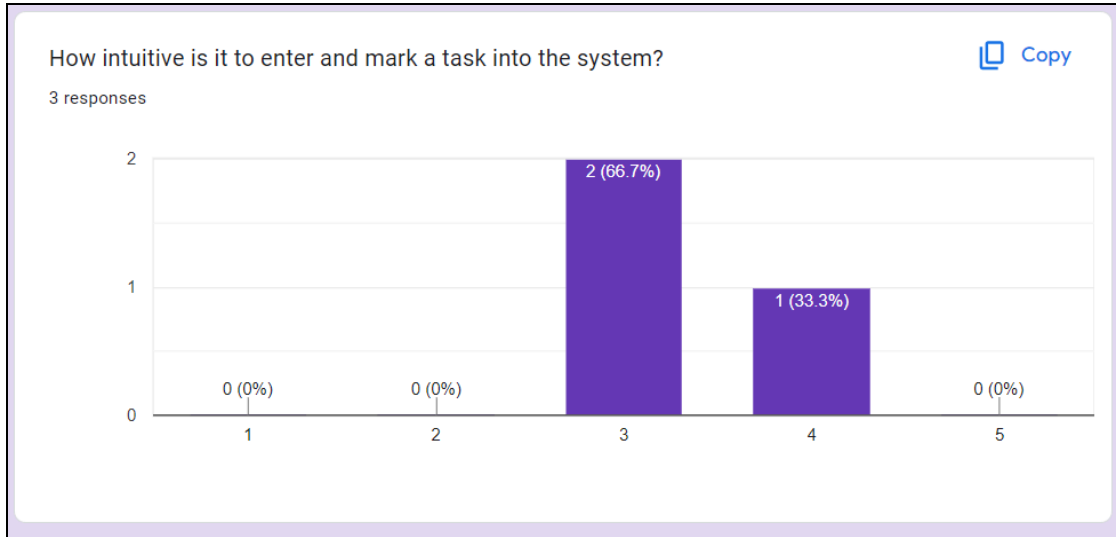


Tasks

The next task for the users was to enter a Task into the system. We asked them to create a random task into the system, mark it as a weekly test and then mark it as done.

We asked them to rate how easy and intuitive it was to enter tasks into the system and to rate the general layout and usability of the screen, the results of which can be seen in figures below

Most participants were confused in regards to what the + icon in the Task meant, they expected it to enter the task into the system but this did not happen. We had to explicitly inform them that they have to press the “ENTER” button on the keyboard to make this happen.



Emergency Contact Page

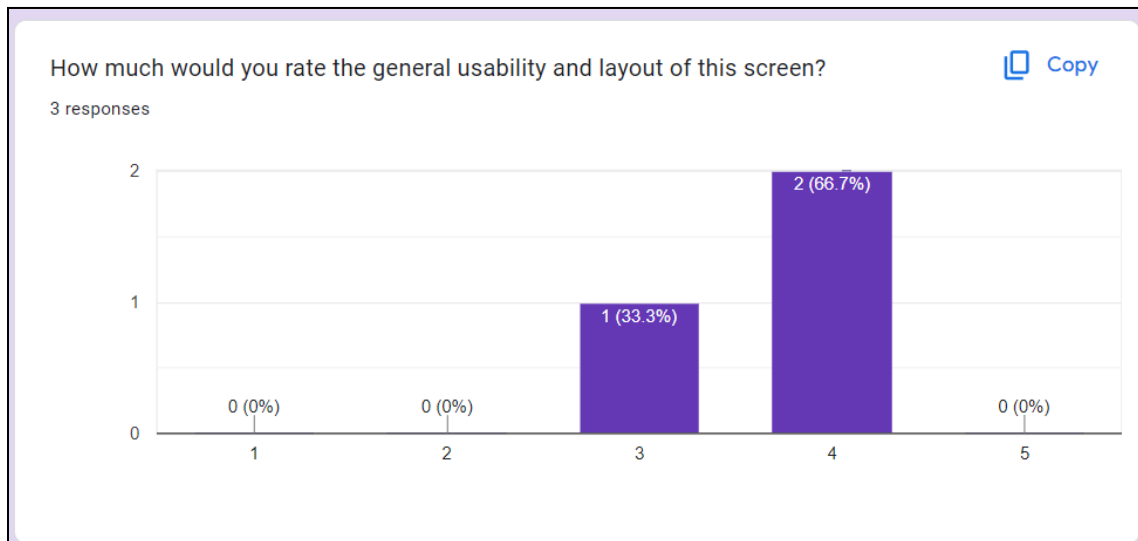
In this section, users were asked to find a contact for an emergency situation.

After using the screen, they were asked to evaluate the general usability and layout of the screen, which is reported in fig

Users also asked the font size and text placement to be improved since it appeared slightly uneven at places.

In general, some users suggested an option to immediately call the number by clicking on an icon on the screen, rather than manually typing this into their phone.

Another improvement suggested was to improve the drop down menu layout which seemed to cover some of the text of the screen.



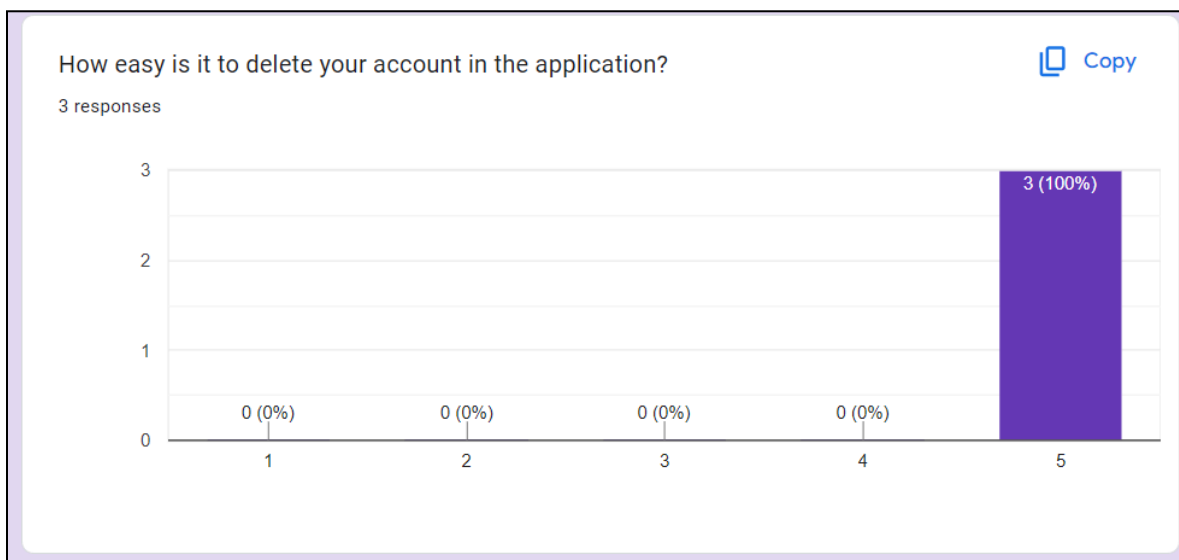
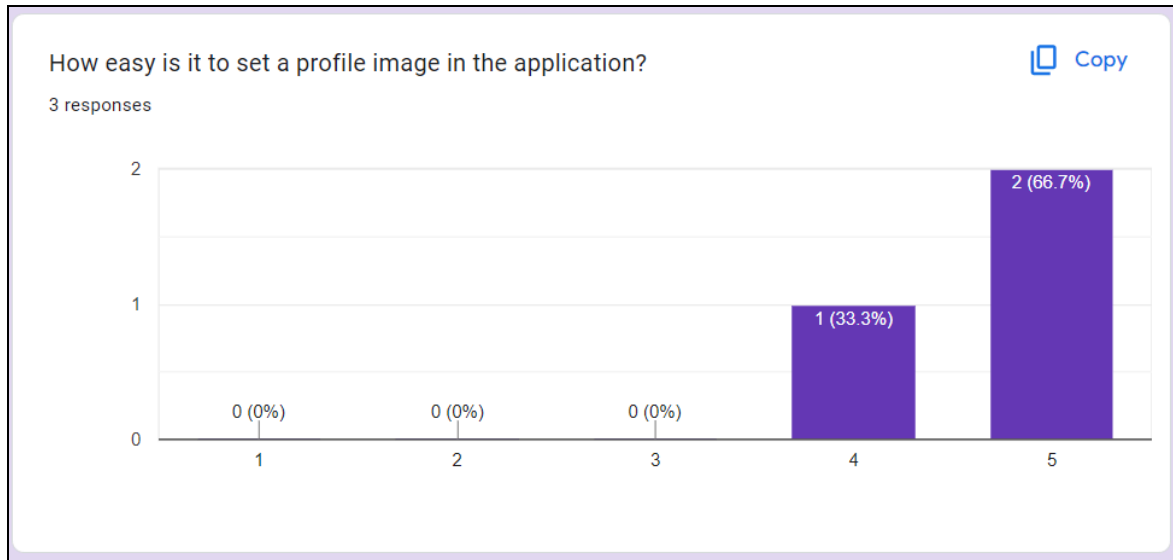
Change Account Details

For this section of the test, we asked the user to change account details. Namely, change the profile image, password and delete their account.

They were then asked to rate how easy it is to change the profile image on the screen.

They were also asked to rate how simple it was to find out how to delete their account.

A common suggestion was that clicking the profile icon on the home screen should also redirect to the edit profile page.



Sleep Tracking Page

Since Fitbit Login is not possible on the Android Emulator, we provided the users with an image of what the sleep screen would look like if they had a Fitbit account and smartwatch.

We wanted to see how intuitive it was to calculate how long a user slept on a certain night based on the displayed graph. This value was 9 hours and was correctly guessed by all participants

A key theme of the screen was to display to the users how their sleeping patterns progress over a period of time. We asked them to rate how easy they found this and the results are shown on figures below

Finally, we asked them to rate the general usability of the screen.

A common suggestion was to also show the hours slept inside the bar of the graph, which currently only shows when the user started and finished sleeping.

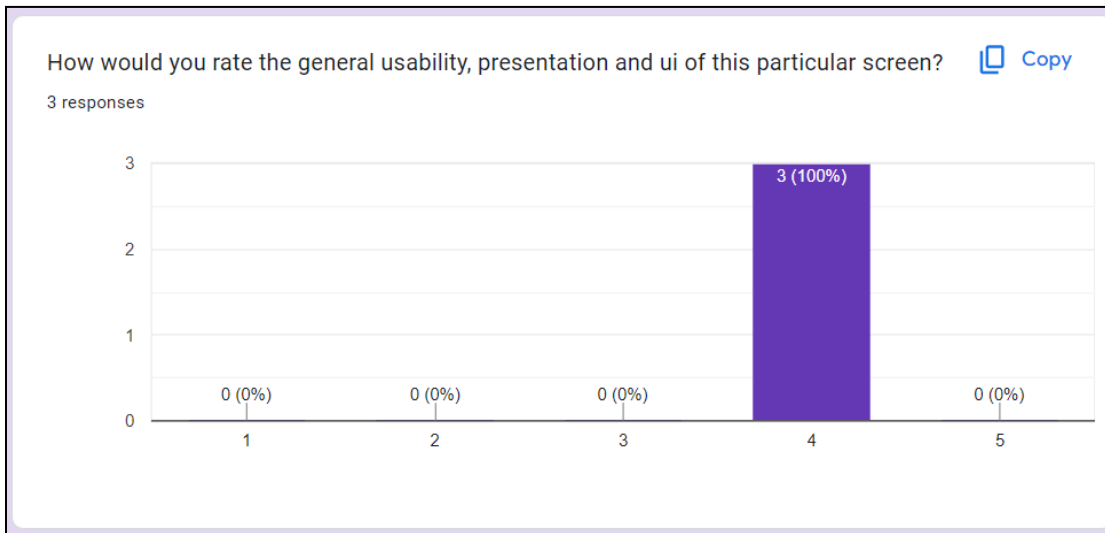
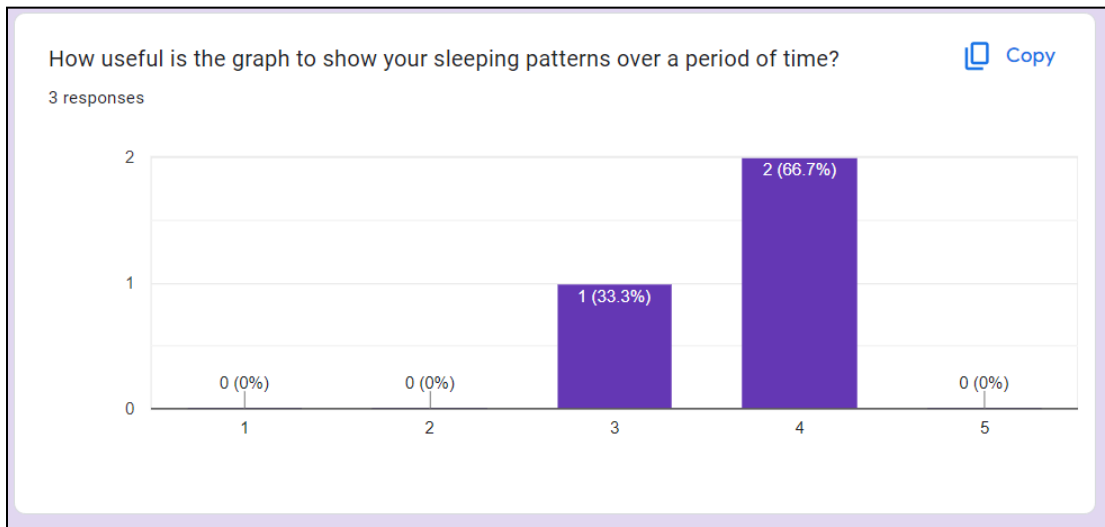
How many hours of sleep did the user get on April 7 according to the graph?

3 responses

Option 1

9h

9

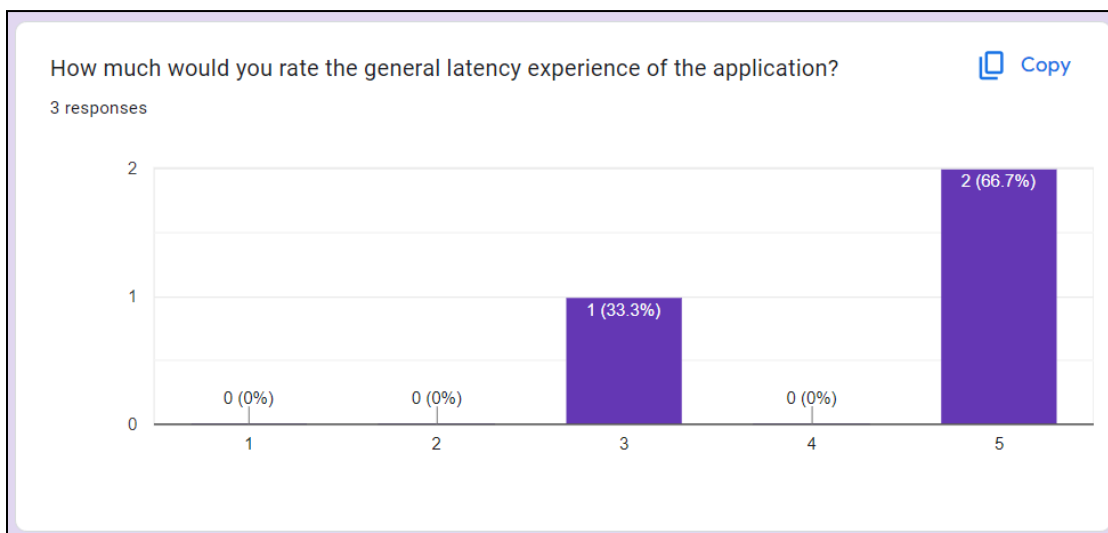
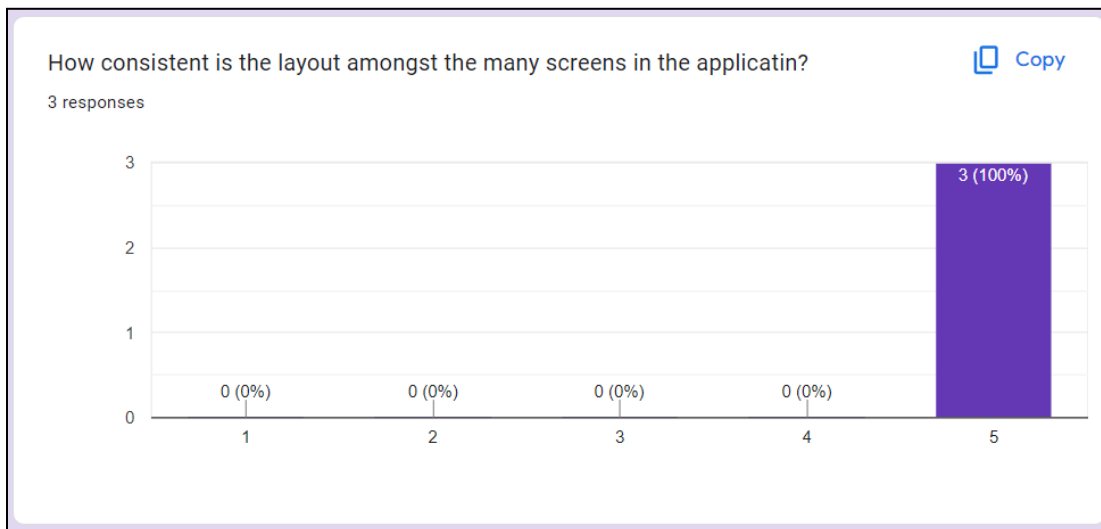


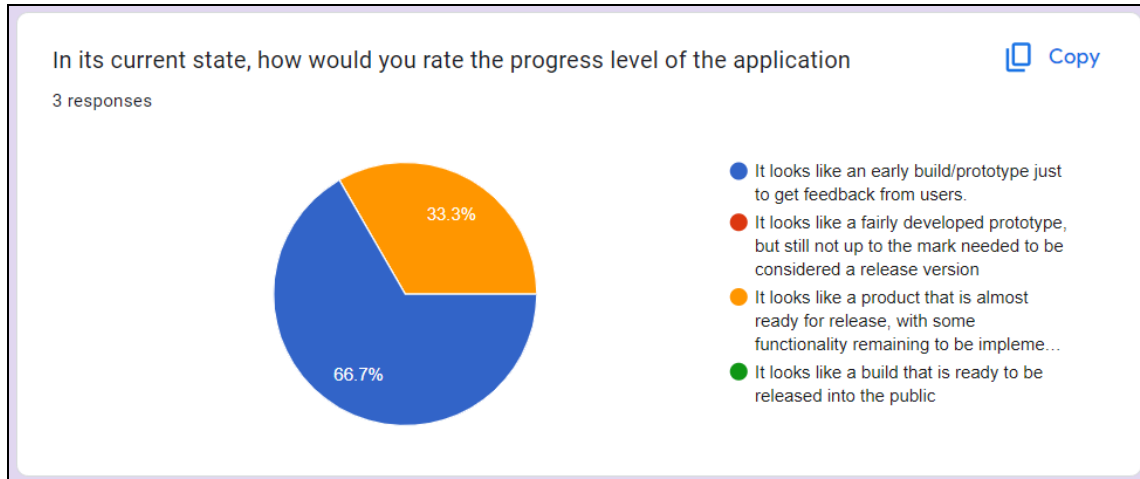
General Usability Questions

One of the development team's primary objectives was to ensure that all the screens in the entire application match the same pattern, even if they are developed by different individuals. We asked the users to rate the general consistency of the themes on all screens.

While the emulator is not a precise indicator of application performance, it is a good indicator of loading times for most tasks and screens. We asked the users to rate the general latency experience of the application.

Furthermore, we finally wanted to ask what stage of the cycle they thought the product was, considering all participants were Computer Science students we believed they would have a good idea if this looked like a ready to release product or an early version of the prototype.



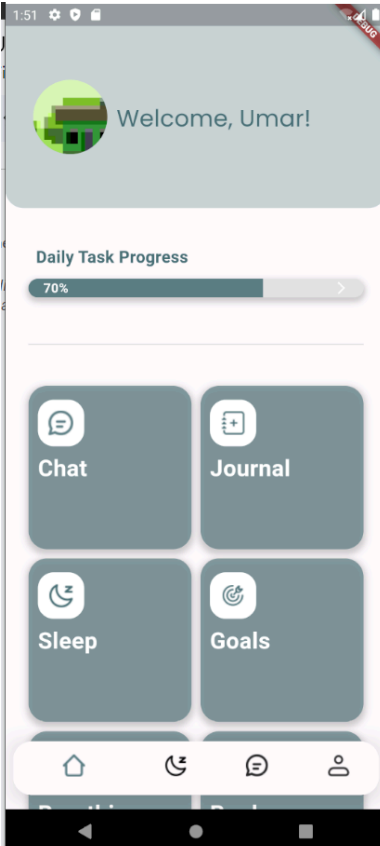
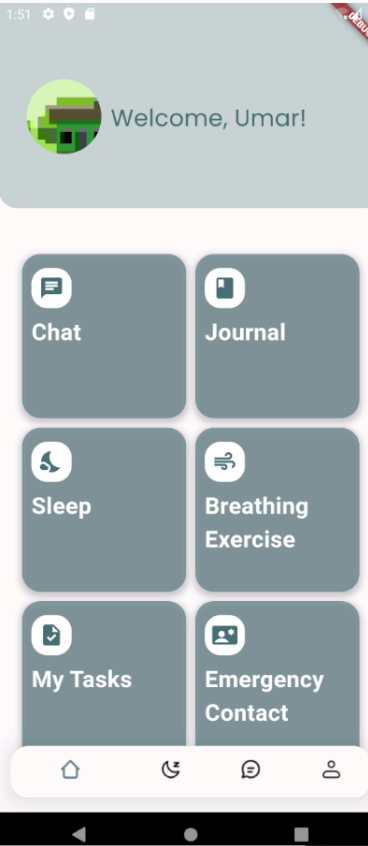


Main Takeaways

- One of the main aspects of future improvements for the application is to allow users to login with their accounts for existing services like Google, Microsoft etc based upon user feedback.
- In general, the feedback for the sign-up and login pages was very positive. It appeared professional, neat and intuitive to use. Very little changes to be made for this section.
- For the home page, we have changed the icons in such a way that none are repeated, as suggested by the users.
- In the future, we would like to add more details to the personalised greeting, since not all users noticed it. It may involve adding effects/animations to make it more appealing and make it stand out.
- Some of the features were worked on, but not fully implemented. This made some users look at the application as a prototype rather than a product ready for release. We removed these from the home screen until they are fully implemented.
- We wish to create more colour schemes for the application, including one with darker options for users that would use the app at night, to ensure that the colouring options are not limiting to users since some did not necessarily like the default option we provide.
- In general, it was difficult to assess the preference of items to place towards the top of the screen therefore the order of tasks will not be changed.
- Journals now have an option to delete entries based upon user request.
- Also based on user suggestions, we now have a colour indicator for the users to input how they feel when making an entry, and this rating can be seen when scrolling through the entries to see general progression of mood.
- As a future improvement, we could offer users the option to either display all or just some of their journal entries. In the current state, however, it only displays the most recent 7 entries.

- In the future, we should implement a redirect from the profile icon on the home screen to the edit profile page.
- Also in the future, we should have a hours slept inside the bar for the sleep tracking graph.
- Breathing Exercises has gone through a major overhaul, with general improvements in the overall layout and usability of the screen.

Design Changes

Screen	Before	After
Home	 <p>The 'Before' screenshot shows a home screen with a grey header containing a profile picture and the text 'Welcome, Umar!'. Below this is a 'Daily Task Progress' section with a green bar at 70%. The main content area contains a 2x2 grid of buttons: 'Chat' (with a speech bubble icon), 'Journal' (with a document icon), 'Sleep' (with a moon icon), and 'Goals' (with a target icon). At the bottom is a navigation bar with icons for Home, Sleep, Chat, and Profile.</p>	 <p>The 'After' screenshot shows a home screen with a grey header containing a profile picture and the text 'Welcome, Umar!'. Below this is a grid of six buttons: 'Chat' (with a speech bubble icon), 'Journal' (with a document icon), 'Sleep' (with a moon icon), 'Breathing Exercise' (with a breathing icon), 'My Tasks' (with a checklist icon), and 'Emergency Contact' (with a person and star icon). At the bottom is a navigation bar with icons for Home, Sleep, Chat, and Profile.</p>



Journal

