Electronics and Computer Science

Faculty of Engineering and Physical Sciences

University of Southampton

Lewis Hawkins - lh5g19

Max Burgess - mwmb1g19

Michelle Moh - mjhm1a18

Peik Lim - pxl1g19

Thomas Hoad - tdh1g19

September 2022 - January 2023

Tracking fatigue after brain injury in real-time using a smart-phone app and sensors/wearables.

Project supervisor: Mark Weal

Second examiner: Mercedes Arguello Casteleiro

A Group Design Project Report submitted for the awards of:

Computer Science MEng
and
Electrical and Electronic Engineering MEng

Abstract

This project created a mobile application designed to help monitor mental fatigue in people who have suffered some form of Traumatic Brain Injury. The core data collection method is through asking participants to complete an in-app survey which has its structure based upon an earlier piece of research by Dr. Leisle Ezekiel aiming to do a similar thing. This project expands upon previously completed work by building an application with more user-friendly features. These include a settings page, notifications and reminders, login systems and a data dashboard where the user can review their history of survey responses. These features are in addition to the optional integration of the application with the Fitbit family of wearable devices and fitness trackers for even more detail.

The project was an overall success and has created an app with a lot of functionality and usability for real world situations. However, there are some openings for future work which are discussed.

Contents

1 Introduction **8**
- 1.1 Problem Statement . . . 8
- 1.2 Solution . . . 8
- 1.3 Collaboration . . . 9
  - 1.3.1 Integration With An Existing Prototype . . . 9
  - 1.3.2 Purpose of New Prototype . . . 9
- 1.4 Product Description . . . 9

2 System and UI Design **11**
- 2.1 Terminology . . . 11
  - 2.1.1 Ecological Momentary Assessment (EMA) . . . 11
  - 2.1.2 Psychomotor Vigilance Test (PVT) . . . 12
- 2.2 Early-Stage Design Process . . . 12
- 2.3 Prototyping on Figma . . . 15
  - 2.3.1 Homepage . . . 15
  - 2.3.2 User survey . . . 16
  - 2.3.3 Reaction Time Test . . . 18
  - 2.3.4 Data Dashboard . . . 19
  - 2.3.5 Settings Page . . . 20

3 Development and Builds **22**
- 3.1 Development Setup . . . 22
  - 3.1.1 IDE and Emulators . . . 22
  - 3.1.2 GitHub Repository . . . 23
  - 3.1.3 Activity Layout . . . 24
- 3.2 User Surveys . . . 26
  - 3.2.1 Survey Development . . . 26
  - 3.2.2 Survey Navigation and Results Collection . . . 26
  - 3.2.3 Response to Design Feedback . . . 27
- 3.3 Reaction Time Tests . . . 28
- 3.4 Notifications . . . 30
  - 3.4.1 Notification Design . . . 31
  - 3.4.2 Notification Persistence . . . 32
  - 3.4.3 Notification Controls in Settings . . . 33
  - 3.4.4 Emergency Notifications . . . 33
- 3.5 Data Dashboard and Daily Summaries . . . 34
- 3.6 FitBit Integration . . . 37
  - 3.6.1 Selection of Device . . . 37

3.6.2 Integration of Device with the App 38
3.6.3 FitBit Display on Dashboard 40
3.7 Storing the Data 42
3.7.1 Room Database 42
3.8 Login Page 43
3.9 Settings Page 45

4 Testing **48**
4.1 Internal Testing 48
4.1.1 Unit Testing 48
4.1.1.1 Facebook Sign-in Problems 49
4.1.2 Integration Testing 50
4.1.2.1 Android Deeplink issues 50
4.1.3 System Testing 52
4.2 External Testing 52
4.2.1 Ethics Approval 52
4.2.2 Setbacks 54
4.2.3 Feedback 55
4.2.3.1 Results 57
4.2.3.2 Analysis 58

5 Project Specification Outcomes **60**
5.1 Technical Objectives 60
5.2 Stretch Objectives 61
5.3 Gantt Chart Comparison 63

6 Project Management **66**
6.1 Project Planning 66
6.1.1 Customer Planning 66
6.1.2 Group Planning 66
6.1.3 Code Planning 66
6.2 Division of Responsibilities 67
6.2.1 Technical Contribution 67
6.2.2 Report Authorship 68
6.2.3 Poster and Presentation Contribution 69
6.3 Risk Assessment Outcomes 69
6.4 GDPR and Ethics Compliance 71

7 Conclusion **73**
7.1 Conclusion of Project 73
7.2 Proposal of Future Work 73

Bibliography **75**

A Original Project Brief **78**

B Electronic Appendix Contents **79**
B.1 Brain Fatigue App Code 79
B.2 Customer Documents 80
B.3 Ethics Documents 80
B.4 Project Management Documents 80
B.5 Survey Documents 81
B.6 System Design Documents 81
B.7 Additional Appendices 81

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of the programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.

We expect you to acknowledge all sources of information (e.g. ideas, algorithms, data) using citations. You must also put quotation marks around any sections of text that you have copied without paraphrasing. If any figures or tables have been taken or modified from another source, you must explain this in the caption and cite the original source.

I have acknowledged all sources and identified any content taken from elsewhere.

If you have used any code (e.g. open-source code), reference designs, or similar resources that have been produced by anyone else, you must list them in the box below. In the report, you must explain what was used and how it relates to the work you have done.

I have not used any resources produced by anyone else.

You can consult with module teaching staff/demonstrators, but you should not show anyone else your work (this includes uploading your work to publicly-accessible repositories e.g. Github, unless expressly permitted by the module leader), or help them to do theirs. For individual assignments, we expect you to work on your own. For group assignments, we expect that you work only with your allocated group. You must get permission in writing from the module teaching staff before you seek outside assistance, e.g. a proofreading service, and declare it here.

I did all the work myself, or with my allocated group, and have not helped anyone else.

We expect that you have not fabricated, modified or distorted any data, evidence, references, experimental results, or other material used or presented in the report. You must clearly describe your experiments and how the results were obtained, and include all data, source code and/or designs (either in the report, or submitted as a separate file) so that your results could be reproduced.

The material in the report is genuine, and I have included all my data/-code/designs.

We expect that you have not previously submitted any part of this work for another assessment. You must get permission in writing from the module teaching staff before re-using any of your previously submitted work for this assessment.

I have not submitted any part of this work for another assessment.

If your work involved research/studies (including surveys) on human participants, their cells or data, or on animals, you must have been granted ethical approval before the work was carried out, and any experiments must have followed these requirements. You must give details of this in the report, and list the ethical approval reference number(s) in the box below.

78456

Chapter 1

Introduction

Firstly, this report introduces the problem statement of the project, the solution proposed to counter the problem, the collaboration with our external customer, Dr Leisle Ezekiel and project supervision under Dr Mark Weal. This is then followed by a general overview of an existing prototype and the requirements of the new prototype set out by our external customer and conclude with a brief insight into the product's description. The body of the report breaks into initial chapters documenting the overall system design of the prototype, describing the development and build of each feature in the prototype. The second half of the report contains chapters which document both internal and external testing of the app, compare the project specification to the end product, an overview of how the project was managed, ending with a conclusion of the project and a proposal of future development for the app.

1.1 Problem Statement

Nearly 400,000 individuals are admitted to hospitals because of brain injury. Acquired Brain Injury (ABI) is classified as an injury sustained by the brain after birth and is usually categorised as traumatic and non-traumatic i.e a fall, tumour or a stroke[28]. The aftereffects of a brain injury vary from one individual to another depending on the severity of the injury. This impacts the normality of their lives, preventing the continuation of work, social and leisure activities. The effects affect mainly their cognitive abilities i.e concentration span, memory retention, slurred speech and etc. However, the common effect of a brain injury found in most patients is fatigue. Managing fatigue and improving the impact it has on daily life is rather difficult due to its nature of being unpredictable and individuals that have deteriorating cognitive abilities may find it difficult to monitor and be more aware of their own fatigue levels.

1.2 Solution

To address this issue, our solution moving forward is to build a mobile app that can aid patients with brain injury, the ability to gain a greater understanding of their fatigue levels and learn how to manage them. This will be done through activity-based surveys and the monitoring of vitals with the help of wearables such as a FitBit or an Apple Watch. Incorporating these features into the app provides a pathway for patients to have more awareness of their energy levels and to monitor their energy levels. The app will then allow them to gain a better understanding of what activities may potentially be affecting their fatigue levels more so than others and prompt the patients to make changes to the activities to improve the impact that fatigue has on their daily lives.

1.3 Collaboration

This project was done in collaboration with Dr Leisle Ezekiel and under the supervision of Dr Mark Weal. Dr Leisle Ezekiel is a registered Occupational Therapist at the University of Southampton in the School of Health Sciences with research centred around social participation after brain injury, experiences of fatigue post-brain injury and ecological momentary assessment of daily activities and fatigue following brain injury or long COVID. Her research also utilises participatory and collaborative approaches such as user-centred design and she is interested in a wider spectrum of mHealth to support self-management of long-term health conditions.

1.3.1 Integration With An Existing Prototype

Prior to the commencement of this project, a working prototype app using an Android operating system was developed and tested by people with brain injuries. This prototype consisted of repeated surveys where people could log their activities and fatigue at intervals throughout the day, it also sampled ambient noise and physical activity transitions, and concluded with a short reaction time test which was an indicator of mental fatigue.

1.3.2 Purpose of New Prototype

A new prototype was needed by the external partner to address design issues and to further the development of the app. Suggestions for the new prototype included the following: collecting physical activity and sleep data from external devices such as a FitBit, providing daily summaries of the responses from the user surveys, and reaction time scores, allowing customisation of the app by the user, providing an enhanced short reaction test to increase user's engagement, tracking of environmental noise, possibly capturing phone use data that may indicate fatigue- a slow response or increased errors on tasks.

1.4 Product Description

As the new prototype is an extension of the previous one, there will be some elements that will remain the same. The prototype will still consist of surveys where the users can report their fatigue using an n-point Numerical

Rating Scaled (NRS), and log their activities by participating in an activity questionnaire which will collect information regarding the user's current/most recent activity and a reaction time test also known as a Psychomotor Vigilance Test (PVT) as a means of assessing any changes in levels of fatigue.

The additional elements include a customisable notification system which prompts the user to take a survey reporting their current levels of fatigue and activity, a dashboard page where users will be able to view a summary of their daily activities which is made up of the responses from the questionnaire i.e the NRS and PVT results, giving the user the ability to synchronise data from their wearables such as a FitBit.

According to our partner's published research, there was an emphasis on allowing more user customisation to the app. The main features of user customisation were centred around allowing users to set the time frame in which they would want to receive the notifications prompts, a time when a daily summary would be ready to view, and allowing the user to switch between dark and normal mode. Last but not least, there is a settings page that housed these user customisation features making it easier for the user to locate.

Chapter 2

System and UI Design

In the previous chapter, we highlighted the need to build an Android mobile application that could help patients with brain injury gain a greater understanding of their day-to-day fatigue levels. As with all software projects, every human-centred product design project starts with the user interface (UI) design process. Considering that the UI would act as the sole point of contact between the user and the app interface, it is crucial that we design an app that balances functionality with ease of use. As we walk through this chapter, the key functionalities of the app will be explained more thoroughly in their individual subsections. This subsection in particular aims to walk the reader through the timeline of the UI design process and additionally, briefly explain the design aspects that were taken into account.

As mentioned in sections 1.3.1 - 1.3.2, a working prototype of the app had already been previously developed by the customer. The previous prototype consisted of several essential features that would allow the user a more personal and individual understanding of their fatigue and the conditions that lead to fatigue. Some examples of these features are the Ecological Momentary Assessment (EMA) and the Psychomotor Vigilance Test (PVT). These features will be explained in further detail in the upcoming sections. We will be implementing these features into our app as well, and the development process will be covered in full in Chapter 3.

2.1 Terminology

2.1.1 Ecological Momentary Assessment (EMA)

An ecological momentary assessment, put simply, is a method of assessing a subject's current experiences and behaviours in the individual's normal/natural environment [6]. This is typically done by repeatedly sampling people's thoughts and activities several times throughout the day, during or close to the time they carry out that activity. The action of repeatedly sampling subjects in their naturalistic environments is especially useful for users to understand and report changes in behaviour, symptoms, and cognitions as they carry out different tasks. The method of collection, whether by using a specific piece of technology or using a diary, may vary [6]. During the early stages of the EMA's first implementation, most EMAs called for the completion of pen-on-paper diaries in reaction to a particular event/target behaviour [17]. However, with recent advancements and the development of portable technology, mobile EMAs have been increasingly popular for collecting data as such due to the many benefits that it offers.

By implementing the EMA on the app, users have a way of logging their behaviours and experiences in the moment and according to automated schedules. Altogether, the increased accessibility and convenience allow users to provide accurate and representative data without having to recall their feelings and actions from a specific time. Because of this, the EMA will be an essential part of our app as it would allow users to track their activities and fatigue levels. The development of the EMA will be covered in Chapter 3.

2.1.2 Psychomotor Vigilance Test (PVT)

A psychomotor vigilance test (sometimes referred to as the psychomotor vigilance task) is a reaction time test that is able to objectively assess the changes of alertness and fatigue of a person. The PVT does this by assessing response times such as the response to a stimulus or information processing times. Depending on the reaction times that are being measured, the PVT is able to pick up lapses in attention that could be brought on by exhaustion or the lack of sleep [10][5]. As the PVT poses as a straightforward and simple reaction time test that can objectively measure brain fatigue, it was very much suitable to be integrated into our mobile application as it aligns with the general purpose of the app.

Typically, a PVT carried out in lab settings would last for a duration of 10 minutes. Within 10 minutes, participants are required to use a handheld device to respond to a visual stimulus. Upon seeing the stimulus, participants would respond by pushing a button on the given device, and the time will be recorded. This would then be repeated several times in 2-10 second intervals until the time is up.

2.2 Early-Stage Design Process

In the early stages of the project, we had the opportunity to meet Dr Leisle to discuss user and app requirements. During this meeting, we were able to view the previous prototype and its features, identify user needs, and most importantly get a brief idea of what the final product will look like. We were also provided with some useful resources that will be used later in the project, such as the user-survey questionnaire that Dr Leisle had designed in her previous user-centred study. To aid the ideation stage of the design process, we took to the drawing board to sketch out the basics of the user interface before deep-diving into prototyping and developing the app. Basic sketches of the app were done digitally on a sketching app on a tablet.

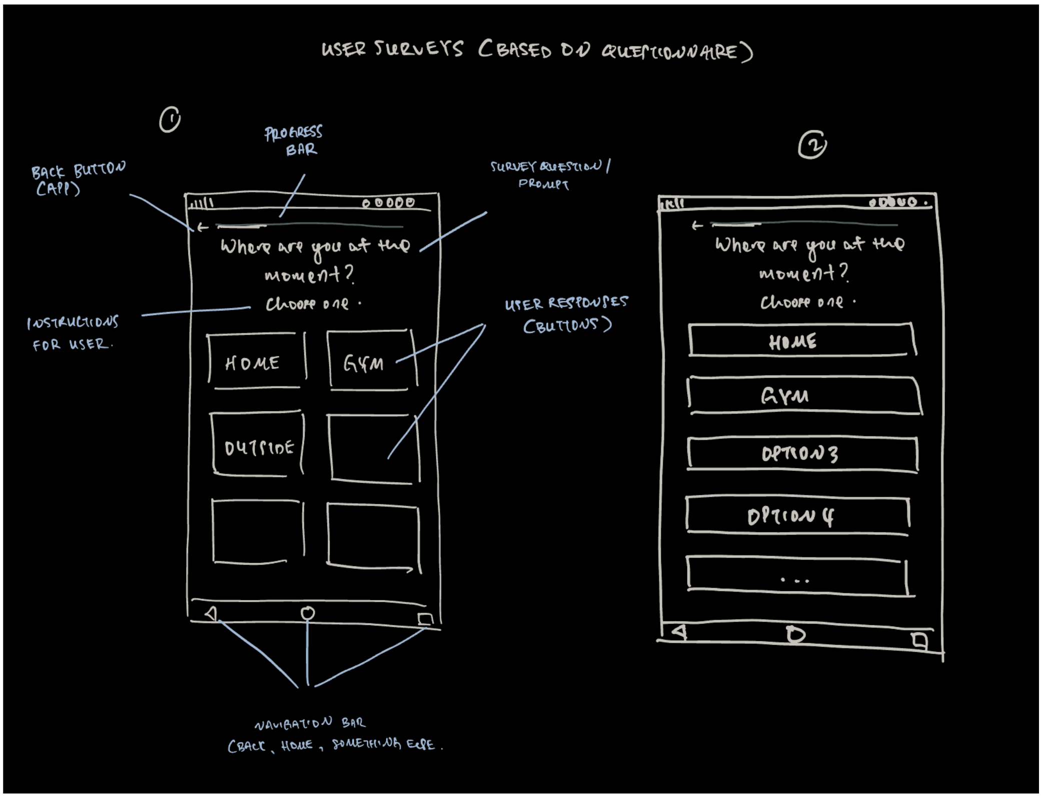

Figure 2.1: A lo-fi sketch of the app's user survey during the very early stages of the project.

Taking a look at some of the examples in Figure 2.1, 2.2, and 2.3 it can be observed that only a minimal number of the visual attributes of the final product were drawn. Of all sketches, the least expressed page/feature at this stage was the dashboard (also called the daily summaries page). The dashboard was ideated to be the go-to page for users to view their responses to their fatigue surveys so that users can monitor and better understand their fatigue levels. At the time, we had an understanding that we will require a feature that would allow users to do so, but we were still in the brainstorming process on how to relay the information back to the user in a simple yet informative way. There is a multitude of ways for us to achieve this, such as using charts/graphs or just by displaying text on a UI card (Android CardView) based on their responses. As this was still in the early stages of the project timeline, we decided to prioritise exploring what types of data we will be able to retrieve such as survey responses and data from a FitBit before designing the dashboard on a higher level. The same applies to the sketch of the PVT (also called the reaction time test) as there were several methods of implementing the test.

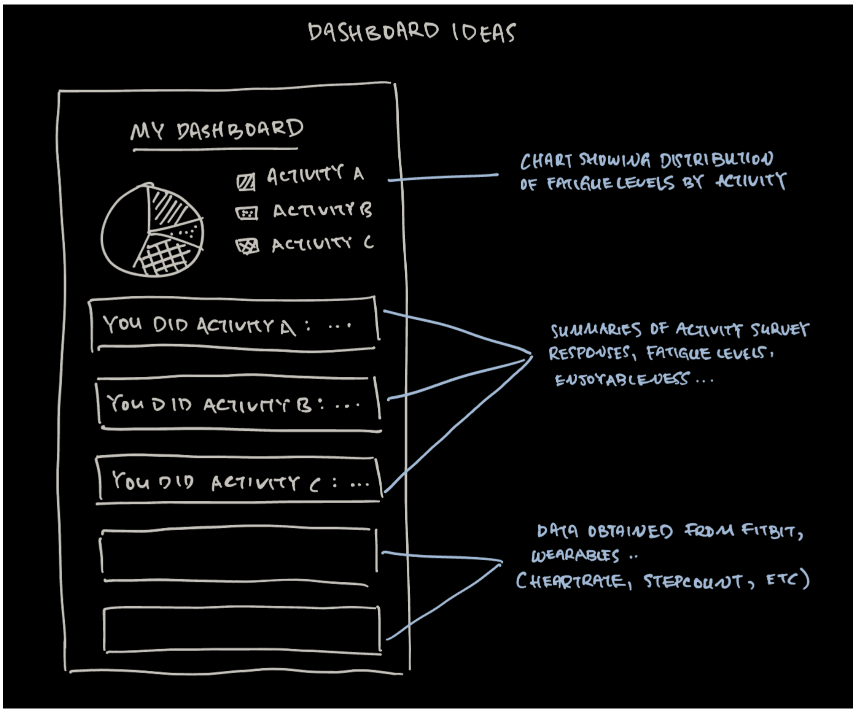

Figure 2.2: A sketch envisioning the look/layout of the app's dashboard page, where users will be able to see their survey responses and other data of the day.

By observing the other sketches, we take note of the key elements of each page, such as the survey question/prompt, instructions for the user, user response and input methods (buttons and sliders), and also navigation and progress bar statuses. The entire process of creating low-fidelity sketches such as the ones shown above only took 10-15 minutes at most and did not require any setup. Being able to draft sketches without too much effort allowed us to quickly switch between methods of implementing certain features, making it extremely advantageous of formulating a clearer expectation of the final product.

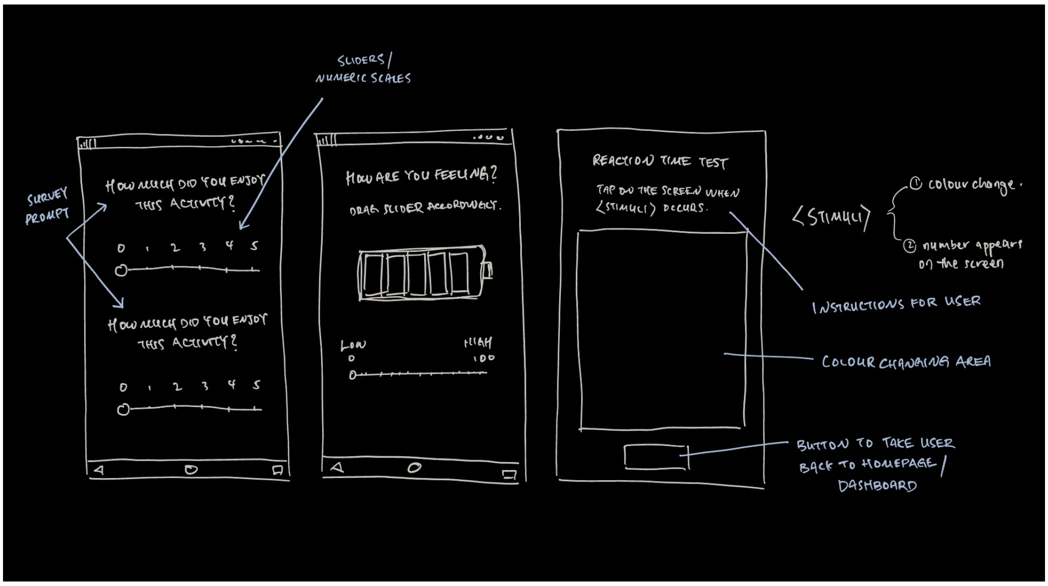

Figure 2.3: Drawings of the app's slider variation of the user survey and of the PVT.

2.3 Prototyping on Figma

After drafting out the sketches, we took to prototyping the app in more detail on Figma. Here we will explain the individual components that make up every page/feature. It might be useful to note that the prototypes shown here may differ from the final product design in terms of the layouts, fonts, font size, navigation buttons and so on. This is the result of continuous improvements being made to the app over time, after getting design feedback from within the team, Dr Leisle and our supervisor Dr Mark. Any design changes will be highlighted in Chapter 3.

2.3.1 Homepage

The initial prototype for the home page of the application included two buttons that will allow users to navigate to two major parts of the app. Intuitively, upon clicking on the first button, the user will be navigated to take the EMA survey; upon clicking on the second button, the user will be brought to the dashboard where they can view their user survey responses. On top of the page, the message "Hello, what would you like to do today?" greets the user and lets the user know what actions he/she can take. The final design largely took on our prototype.

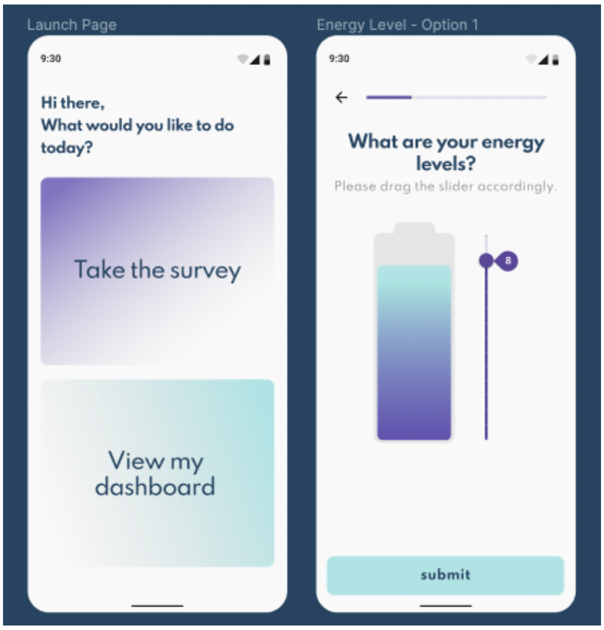

Figure 2.4: Figma prototype of the homepage and battery slider question of the survey.

2.3.2 User survey

The user survey part of the app will collect user responses mainly via two input methods: sliders for numeric rating scales (i.e. 0-10 or 0-100 point scales) and buttons to input user's selections to particular questions. In the prototype, the first part of the survey as shown in Figure 2.4 consists of a question asking the user what their energy levels are. Instructions are given to the user to drag a numeric pointer scale slider (between 0-100) according to how energised they are feeling. Above the slider, a picture of a battery is included as a visual aid. A progress bar is also shown on the top of the page to give the user an idea of how much of the survey is left. At the bottom of the page, a next button is included to allow the user to navigate to the next question of the survey.

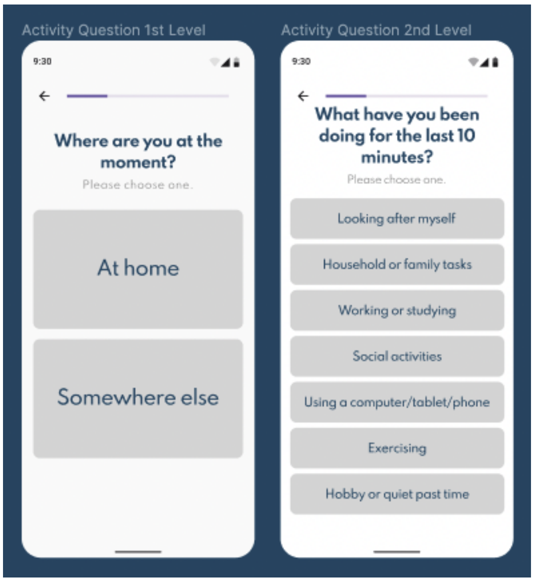

Figure 2.5: Figma prototype of the app's user survey. Vertically stacked buttons are used to maximise screen real estate for questions that have more options to choose from.

In this part of the survey, we ask questions about the user's current or recent activity (an activity that was done in the last 5 minutes or so). The questions and answers we used in the survey had been given to us by Dr Leisle and had already been designed to capture as accurately the recent activities of a user, without the need of asking too many questions. This is firstly done by inquiring about the user's current location (e.g. at home or somewhere else). After that, a more general question follows, asking about the broader category of the activity they are engaged in (e.g. social activities or working and studying). Finally, we get into the specifics of the activity the user is carrying out (e.g. cooking or buying groceries). By doing it structurally this way, we eliminate the need to include the hundreds of activities that a person could be doing. This prevents users from getting overwhelmed by the number of possible selections and keeps the survey simple. From the screenshots of the prototype above, buttons are once again used here as the main method of collecting user responses.

Moving on, the last set of questions of the survey are specific to the activity that the user just carried out. Questions like how much effort it takes, how have your energy levels been affected, and how fatigued are you, are prompted to the user. Depending on the question, sliders and buttons are used to collect user input data just like the previous pages that we have implemented in the prototype. The same elements that were mentioned before, such as the progress bar, back button, and next button, are shown on each page in the prototype. After answering the final question, the user should be able to click the "submit" button on the bottom part of the screen to end the survey. The final design can be seen in Section 3.2.3.

2.3.3 Reaction Time Test

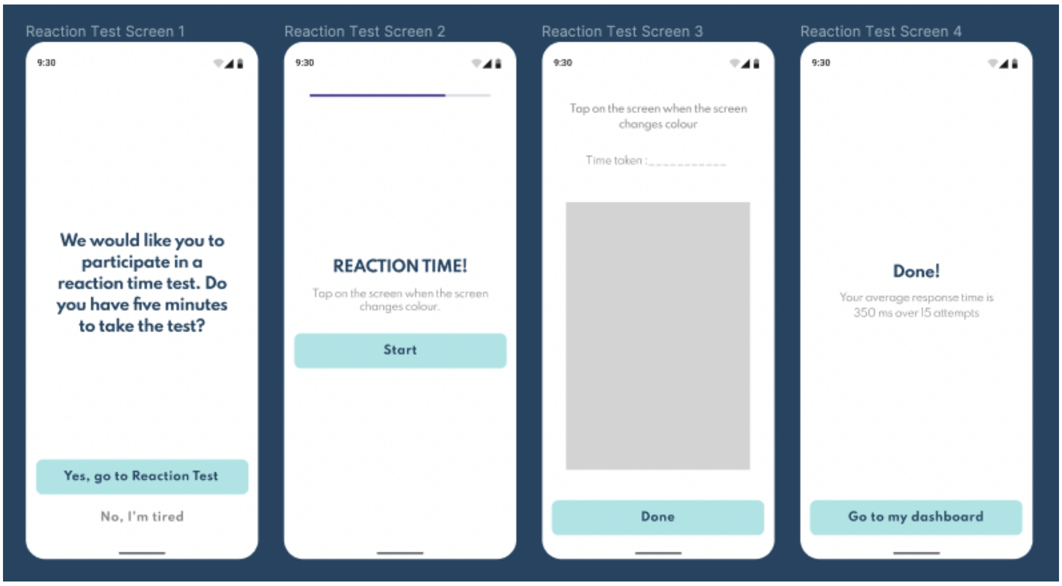

Figure 2.6: Figma prototype of the app's PVT.

After a user has completed the survey questions, they will be invited to participate in a reaction time test. The Figma prototype of the reaction time test only consists of four screens, all of which are static images and do not showcase how the test works. Taking a look at the screenshot above, the first screen contains some prompts inviting the user to take the reaction time test. The user has two choices here, either to go to the reaction time test or to skip it. If the user chooses to skip the test, they shall be navigated back to the homepage. If the user chooses to take the test, then they are navigated to the second screen (see figure 2.6), where a prompt provides instructions to the user to tap on the screen when it changes colour. A start button is included below the prompt for the user to start the test when they are ready. When the user taps on start, they move onto the third screen, where the actual reaction time test takes place. In the centre of the screen, the prototype shows a grey area – this area would be the part of the screen that will undergo a colour change. It is also the area where the user is meant to tap when the stimulus occurs.

Not pictured in the prototype are the actual mechanics of the reaction time test, where each colour-change-then-tap cycle will be repeated for a specific duration. More of this will be further discussed in Chapter 3. In the prototyping stage, we also aimed to display the user response times after each iteration, hence it is included in the prototype. Once finished with the test, the user can click on done and head to the last page (i.e. the fourth screen) of the reaction time test, where the average response time is displayed. From there, the user can navigate to the dashboard to view their survey responses and reaction time test scores.

2.3.4 Data Dashboard

Another vital section of the application is the data dashboard – an area of the app where the user is able to see their responses to fatigue surveys and have their data visually displayed to them. It was important for us to include a data dashboard in the app so that the users could review the history of their responses, with the aim being that they could use the dashboard to monitor and better understand the causes of their fatigue.

We had multiple goals when designing this section of the application. Firstly, we needed the data dashboard to show daily summaries of the user's fatigue in addition to the full data collected in the time since the user first began to record responses to surveys. While the daily summaries allow the user to reflect on their recent fatigue at the end of the day, we also want to provide users with the ability to look back over a larger time frame and at specific dates and times to view responses they may have otherwise forgotten about. We would also aim to separate these sections clearly in the name of creating a simple experience for the user. Furthermore, we had the goal that both of the previously mentioned parts of the data dashboard would display data related to the user's energy levels, reaction time and wearables in a visual manner that is not cluttered or confusing to the user. This visual presentation of the various data related to fatigue will aid users in spotting any trends or patterns in their fatigue over time which could lead to them being able to better manage or control how they experience their fatigue. The data should be shown in a helpful way in order to allow users to achieve this as easily as possible. Similarly, we had the goal that the user should be able to view the full data from every time they answered a fatigue survey including the date and time that the survey was taken and the individual response to each survey question. Once again this was to be done in a way that is intuitive and helpful to the user and that does not cause any confusion itself. This would further aid the user in analysing the causes of their fatigue by reminding them how they answered surveys from a time that they may not otherwise remember.

The solution that we designed uses a horizontal bar with two tabs to allow the user to select to view the 'daily summary' or the 'lifetime' section. As the user does not need to be able to see both of these sections simultaneously, we are able to use the remaining space on the screen below the tabs to display the contents of the user's selected section. We then decided that this space would be divided vertically into an area for graphs of different data and an area for a list of the user's survey responses.

The top of each section would be home to an area for a series of charts that would provide a visualisation of the user's fatigue data. This graph section would be home to three different line graphs – one for the user's energy level (Question 1) in each survey response, one for the user's reaction time result and one for data collected from a user's wearable, specifically a FitBit. Each of the graphs would individually take up the width of the screen with the user having the ability to scroll horizontally to change the graph that would be shown. Furthermore, each of the graphs would be given a unique colour to assist the user in not getting lost or confused when navigating between the graphs as they get more familiar with the application.

We decided that the remaining section beneath the graphs is where the user will be able to view the data from previously answered surveys. For the 'daily summary' section the user will only be able to see their response to surveys taken on the date of the most recent full day whereas for the 'lifetime' section the user will be able to see details of every survey they have ever taken. In both cases the user will be presented with a vertical list of shortened versions of each survey response, sorted by the date and time that the survey was taken. The user will be able to scroll up and down through these briefs before selecting a response to view the full data in a popup.

Our initial proposed solution for the dashboard, as presented in our progress report on the 16th of November, included a single graph at the top of the screen with the tabs and separate sections below that. We moved on from this solution when we confirmed that we would be presenting multiple types of data visually. Furthermore, this does not allow for a clear enough separation between 'daily summary' and 'lifetime' data graphs. While we could also implement the solution from the final implementation here, with six individual graphs to scroll between, this would not meet our goal of presenting the data in a way that is not confusing. Other alternative solutions for the dashboard included having just a single graph to display the three different types of data in each section. We decided against this approach as it would not have met our goal of creating an uncluttered display. Plotting more than two types of data on the same line graph introduces a difficulty with drawing scales on the graph axes and therefore while having a collection of lines on the same pair of axes may have made identifying some trends over time somewhat easier, we decided in favour of separating the data into separate individual graphs with their own colours and scales.

2.3.5 Settings Page

We also decided to include a settings page in our application as we felt that it was important that the user is able to customise their experience with the application, including giving the user some control over the timing and frequency of the notifications that are sent to the user's phone to remind them to take a fatigue survey. Furthermore, we decided that the user should be able to block out certain times of the day when they do not want to receive notifications, such as when they are working. These features were crucial to include in the application because they allow users to only receive notifications at times when they are able to take the survey and are therefore happy to log their responses multiple times every day. Additionally, we wanted the settings page to contain the normal features and adhere to the normal conventions that are commonly found in settings pages, in order to increase the user's ability to use the app with ease and without confusion.

Following Android's own settings design guidelines [19], we made sure that our design for the settings page contained items that are ordered intuitively with a clear title. The guidelines also influenced our decision to place the the user's notification controls at the top of the page as this is where Android recommends placing frequently used settings. The solution we designed would incorporate the use of a drop-down menu followed by a series of sliders to present the user with a clear method to adjust their notification preferences. This would include a slider to specify the earliest and latest time of the day between which they would like to receive alerts, a 'do not disturb' slider which would prevent the user from receiving alerts between two specified times and a slider to let the user set the time of day when they receive a notification to inform them that their new daily summary is available to view in the dashboard. These settings would be placed in a 'notification preferences' section at the top of the settings page and would precede the page's remaining contents, including an FAQs section and a button to let the user log out of their account.

Each slider would be clearly labelled to communicate to the user the changes they are able to make, and any other buttons would also be given a text label and a small icon to increase the user's familiarity with the app. Moreover, the 'do not disturb' slider would be made red to ensure that the user understands that they would *not* be receiving notifications during this period. Similarly, the log-out button would also be red and would be the very final button located at the end of the page as is commonly found in many other applications.

Chapter 3

Development and Builds

3.1 Development Setup

The main aim of our development was to get started with a basic version of the app that could run user surveys. The surveys created a large number of dependencies on other features we wanted to implement and this basic template was important to establish quickly. We went in without any experience in app development so it was critical we came up with an approach that we could all get comfortable with quite quickly.

3.1.1 IDE and Emulators

To develop the app, we decided on using Android Studio with Java. We discovered that IntelliJ had Android Studio integration so we used this as a platform that we were familiar with, provided Git controls and would also provide all the Android Studio features. The app development provided us with a template app to get started with which meant we could immediately start working on features right out of the gate. Most of us are very strong with Java, so even though Kotlin is quite a popular app development language and was an option for us, we decided to go the familiar route.

Shown below in figure 3.1 is a look at our development environment. Here we can see an example of how we are using the Android Studio tools such as setting up an emulator to test our app with ease. We decided collectively on using the same emulated device so that our displays would all look the same as each other. The device we simulated was a Pixel 5 running Android 11, API 30. We chose these specifications because API 30 is compatible with a solid 40 per cent of devices and provided us with essential tools that were required for notifications. We considered using a device running Android 8 or 9 to give us compatibility with 90 to 75 per cent of devices but the overall UI look wasn't as clean and made notifications significantly more challenging to display consistently.

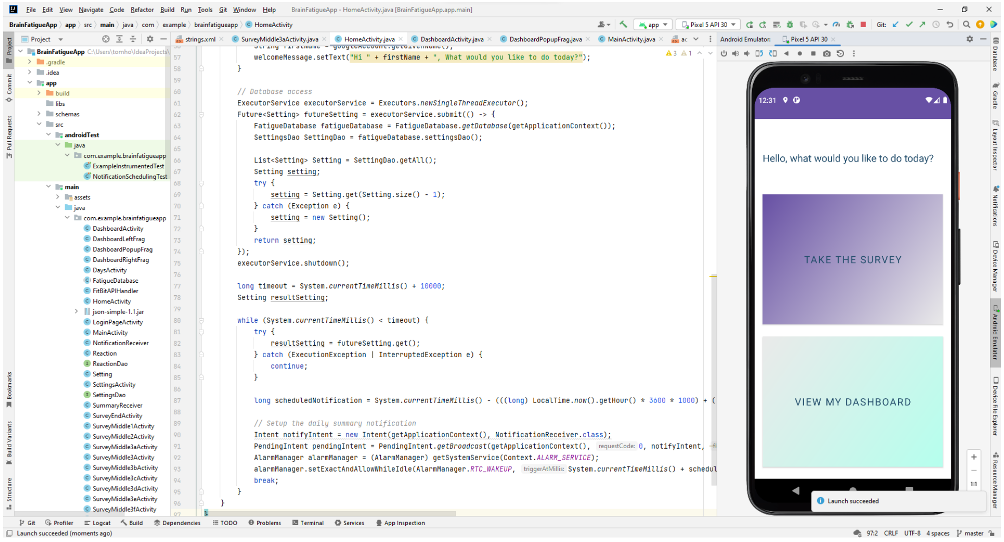

Figure 3.1: IntelliJ Development Environment and Emulator

In addition to the Emulator setup within IntelliJ, we also had a couple of our own Android devices that we used for additional testing during development. Setting up the app on these devices was very easy. Android provides development settings that allowed us to download our app from IntelliJ to the device over WiFi and it became fully functional. This was very beneficial for getting a proper hands-on feel of the app, as well as allowing us to test notifications being sent without having to leave an emulator and hence a computer on overnight.

3.1.2 GitHub Repository

To work collaboratively, we set up a private GitHub to store our work. We used the methodology of creating feature branches for each idea, creating commits on this branch, and then creating a pull request when ready to merge into the master branch. During development, we set up these pull requests to also squash the commits before a merge which also formalised our Git history.

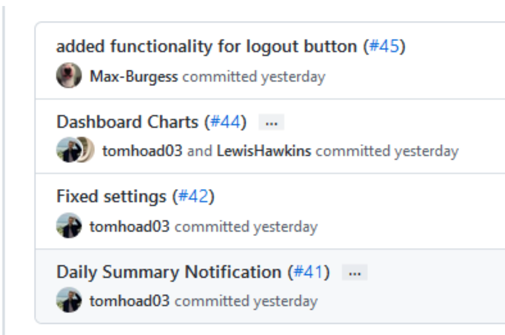

Figure 3.2: Git Pull Request History

We encountered a few problems with Git throughout development, but most issues were quite resolvable using IntelliJ's useful Git integration features. Using pull requests rather than all working on master meant we had very few conflicts arising and meant we always had a branch with a working product we could demonstrate or test with at any given time.

3.1.3 Activity Layout

The main principle of using Android Studio is developing the app using activities and fragments. Activities act as a new page whereas fragments can be used to display more space on a single activity. Through our development, we set up a template using mostly activities, with the use of fragments within the dashboard. Shown below in figure 3.3 is the layout of activities that make up our app.

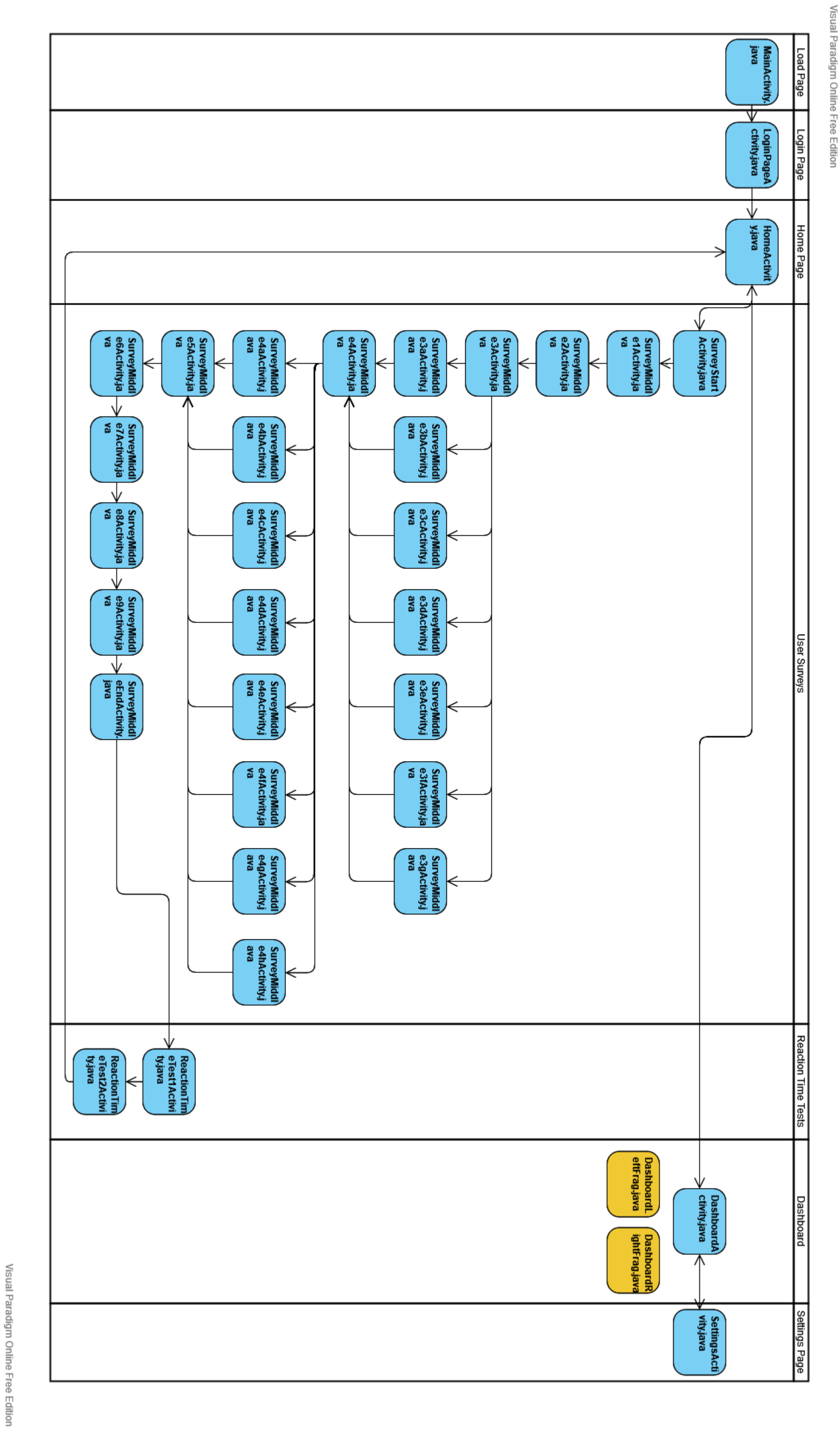

Figure 3.3: Activities Layout

3.2 User Surveys

3.2.1 Survey Development

The development of our survey largely took on our UI design and question structure given in the preparation. We had to make a few changes here or there from our design for elements that didn't look right when translated to a proper device and we went through several iterations of the design.

The user surveys were the first part of our development so, a lot of the UI development was brand new to us and took a while to adjust. With a lot of elements on the page, we had a tough time making sure they were arranged correctly and that layout translated to an actual device. This task was done collaboratively and over the entire duration of the project as we got more comfortable with our development skills and learnt new tricks. Incremental feedback was useful to make gradual improvements over time that ultimately helped us match up the surveys to our initial design.

Shown below in figure 3.4 is an image during the early stages of user survey development. At this point there are only general layout features added, not all questions are fleshed out and the look of the design is only half done.

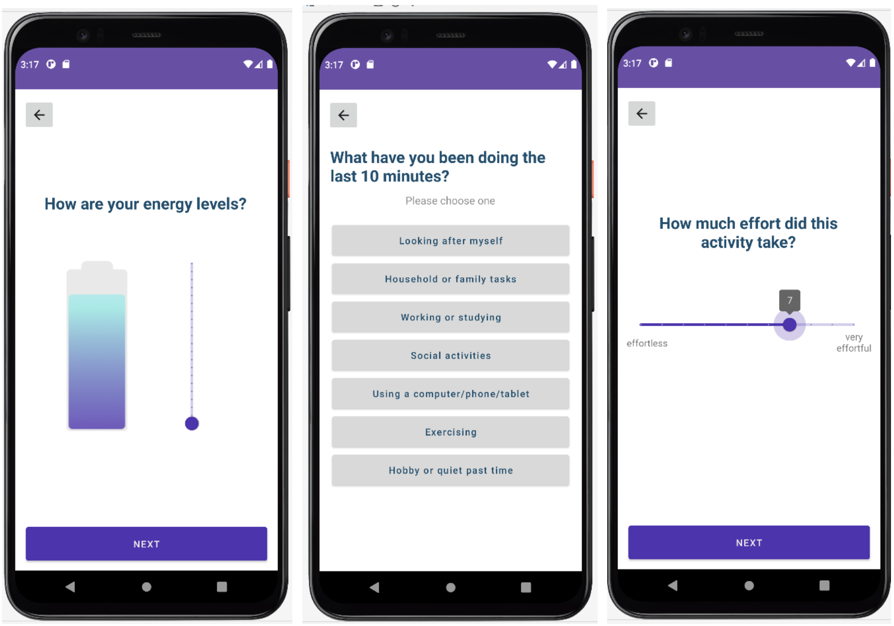

Figure 3.4: Early User Survey Pages

3.2.2 Survey Navigation and Results Collection

Once we had the initial development completed, we needed to get some kind of survey result collection implemented. Whilst perfecting the UI was important, we also needed to make sure the data was available for other parts of the app such as the daily summary and the lifetime dashboard. To collect the information we created a Java class called *SurveyResult.java* shown in appendix B.1. We used the activities chained together to create a *SurveyResult* object with the values inside to reflect the button clicked on. Navigating back in the survey made sure to properly update the results. When the end of the survey is reached, the object is saved to the app database and marked with a timestamp.

Android development uses intents to navigate apps which directly works with Android devices' built-in home and back button. During the initial stage of development, we designed our back button for the survey which we discovered was a bad course of action. Referring to this article [26] we discovered that it's a waste of space and also messes with the stack tracking inherent to Android. The screen real estate we gained from the decision to remove the button was very beneficial and we used it to make improvements later on. An important and final part of our user survey design navigation design was to program the app to block off returning to the user survey once it has been submitted. This prevents a user from repeatedly submitting the same survey and confusing the results.

3.2.3 Response to Design Feedback

During a customer-supervisor meeting we had at the beginning of development, we went through the user survey design and received feedback to improve the UI. These changes were largely implemented in the final version of the design. We also came up with a few improvements of our own to make the user experience better. These changes include:

- Improved question wording and order.
- Improving slider variety and scaling to give numbers and a key.
- Cleanly attaches the reaction time tests to the end of the survey.
- Removing our custom back buttons, replacing them with a survey progress bar.
- Implementing our custom colour scheme to provide a more comfortable user experience.

Shown below in figure 3.5 are a few examples from our final user survey design:

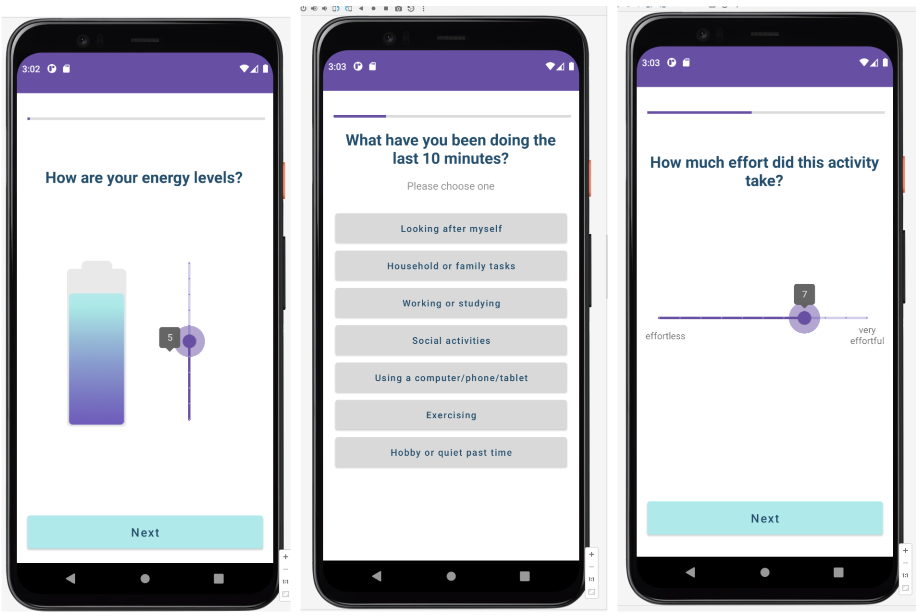

Figure 3.5: Final User Survey Pages

3.3 Reaction Time Tests

After most of the user survey development had been completed, we set off to begin the development of the in-app PVT/RT test. At this point, we were much more comfortable with developing in Android Studio. Because of this, the PVT was able to be completed without us bumping into any major issues, although there were still some minor ones that set us back a couple of days.

Before getting started with the development process and jumping into code, we had to discuss and agree on the requirements of the PVT as there were several things to consider, such as the type of stimulus to present to the user and the duration of each test. Typically, PVTs that take place in lab settings often last for a duration of 10 minutes. Although this was considered the gold standard of PVTs, it was fairly unrealistic to carry out in real-world settings [3]. Considering how users will be expected to take the EMA user survey up to 3-5 times a day if the test lasts 10 minutes each, users could be spending up to an hour of screen time on the app every day. The total screen time may add up even more if we factor in the time it takes to complete the survey and also the time spent viewing the dashboard. This is something that we would prefer to avoid, as an important part of the app is not to cause disturbances to a person's regular schedule or routine. Therefore, we prioritised keeping the PVT short, but not too short that it will affect its ability as an objective measure of fatigue.

Although the ongoing research on PVTs is far from being definitive/conclusive, some studies suggest that a 2-3 minute PVT seems to strike a balance in terms of time and effectiveness in spotting any lapses in response times. In a study conducted by Loh in [3], the effects of sleep deprivation on reaction speeds in the first 2-5 minutes of a 10-minute test were compared to the effects of a full 10-minute test. In the study, although the sensitivity of the test to detect the effects of sleep loss increased with test length, changes in reaction times were noticeable in the 2-minute test [3]. Another study conducted by Basner and Rubinstein in [13] developed a 3-minute version of the PVT. Basner et al noted a 21% decrease in the PVT's sensitivity in detecting changes to alertness brought on by insufficient sleep. However, it was brought to their attention that the advantage of a shorter PVT exceeded the loss of sensitivity [13]. It is because of this that we decided to design a PVT that lasts for 3 minutes. We certainly acknowledge the limitations of a shorter PVT in lab settings, therefore we only suggest utilising the in-app version of the PVT to measure changes in personal performance instead of comparing across persons.

In terms of the mechanics of the PVT, we agreed on using a colour change to be the visual stimulus that will be presented to the user. This colour change will be presented in inter-stimulus intervals (ISI) of 2-10 seconds as defined in a study conducted by Basner and Dinges [12]. The ISI is simply the time interval between the previous response and the onset of the subsequent stimulus [12]. The steps can be summarised as follows:

1. User starts the test, and a 3-minute timer starts.
2. User waits for stimulus.
3. Stimulus appears on the screen in 2-10 seconds.
4. User taps on the screen and response time are displayed.
5. Repeat steps 2-4 until the 3-minute timer is up.
6. Reaction time test ends, and the average response time is displayed and saved to the database.

After deciding on the general criteria for the PVT, we had to think of the design of the test such that it could be fitted into a single page of the app. To do this, we needed 2 buttons for the PVT, one to start/end the test, and one for the user to tap on when the colour of it changes. We decided that giving the user a larger area to tap on the screen would be more appropriate. This way, they can focus purely on performing to their best ability without having to worry about tapping on an exact spot on the screen, which would otherwise shift their attention as a result. When coding the PVT, getting the colour to change at random 2-10 second intervals was fairly straightforward. However, we did face some minor challenges when trying to get the program to loop for a certain amount of time, but this was resolved quickly.

Looking back at the prototype that we had designed for the PVT in chapter 2, most of the key components that were in the prototype were brought to life in the final design of the app. The differences between the prototype and the final design (other than the prompts/font layouts) are summarised below:

- Instead of having four separate pages dedicated to the PVT, we have reduced it to two.
- The start button has been moved to the page where the PVT takes place.
- User's response times will be displayed in the middle of the colour-changing area after each iteration, instead of at the top of the page.
- After the test ends, the app will compute the average response time and display it in the middle of the colour-changing area.
- The start button also changes to a back-to-homepage button when the test ends.

The final design of the PVT can be seen below in Figure 3.6.

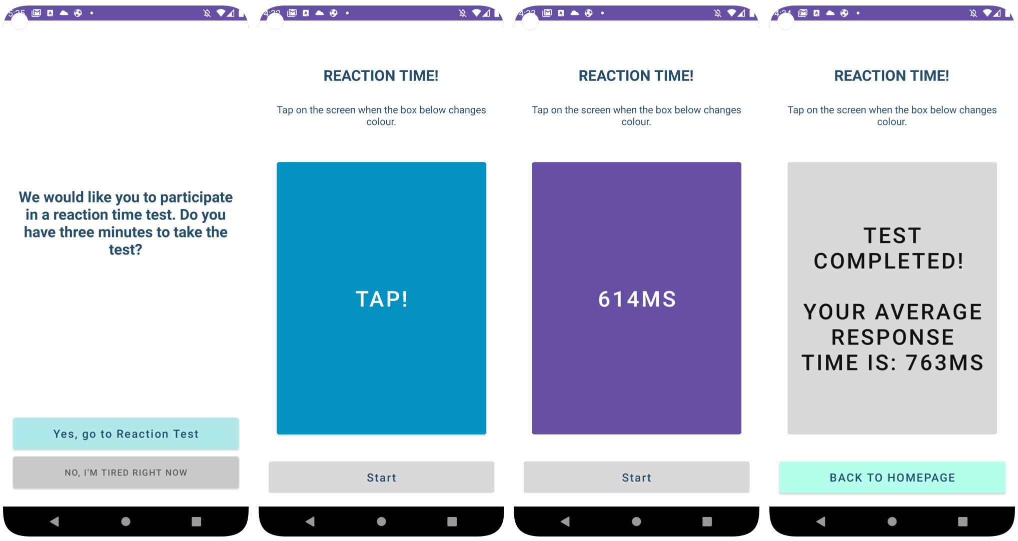

Figure 3.6: The final design of the reaction time test. The user's response time is displayed after each iteration. When the test ends, the user's average response time is also displayed.

3.4 Notifications

The notifications were a critical feature for the app and were perhaps the hardest part to design well. It took us a good length of time to get these systems in place correctly and they ended up taking longer than we had planned in our Gantt chart. This was partly because of its dependency on the settings page which was designed late on, but also because Google is continually scaling down how certain background functionality works in Android apps.

We have multiple different notifications for this app. Firstly, we have a brain fatigue survey notification that goes off a number of times a day at fairly regular intervals. Secondly, we have a daily summary notification that will give users a rundown of their surveys in the last 24 hours. Finally, we have a second survey notification that is designed to go off in extreme scenarios such as too much noise, too much movement or a large change in fitbit data.

3.4.1 Notification Design

The notification design and building all happen within one Java class, NotificationReciever.java which is highlighted in appendix B.1. Within this class, we lay out the design features for a notification. We also use another receiver to send the summary notification which has a very similar design to this. The following features apply to all notifications:

- Vibrate when notification is received.
- Forces the notification to appear as a heads-up display.
- Categorises the notification as a phone call, increasing general importance.
- Sets up the notification to launch the app to a specific activity when interacted with.
- Sends the notification even if others are suppressed.
- Dismisses the notification when the user interacts with it.
- Prevents the user from dismissing the notification any other way than interacting with it.

As a result, this provides us with a notification that will always persist on the screen for the user, meaning they have to interact and start the brain fatigue survey to remove it. We decided on this design choice because we placed high importance on these notifications, and we wanted them to get a guaranteed response from the user. It also meant that implementing noise response notifications would allow for an immediate interruption. As a result, this is how the notification will appear to the user.

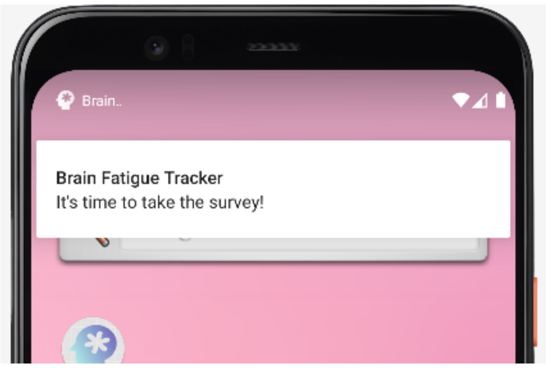

Figure 3.7: Fatigue Survey Example Notification

3.4.2 Notification Persistence

A key part of the survey notifications is to make sure that they are seen by the user and that they will act upon it. This was quite challenging to achieve, especially as Google is in general cutting back techniques for processes that run in the background. These are efforts aimed to conserve battery life as much as possible but this meant we had to work around several restrictions. Whilst all these changes seem to have been implemented in newer versions of Android, we needed to work with them rather than avoiding them by working in an out-of-date version. We got quite creative with our notifications system and ultimately it gave us a lot of flexibility.

Survey notifications should be regular, semi-predictable and customisable. We managed to achieve this by calculating when each notification was going to occur after every completion of the survey. The phone looks at the current time, the available times and the desired frequency of notification and works out the next available time to send the location. We alert the user to when the next notification aims to arrive so that they can get a general idea of what time it will arrive. Given Android's sleep measures, we noticed there was some variance from this calculated time which we appreciated in some regards and became quite annoyed at in others.

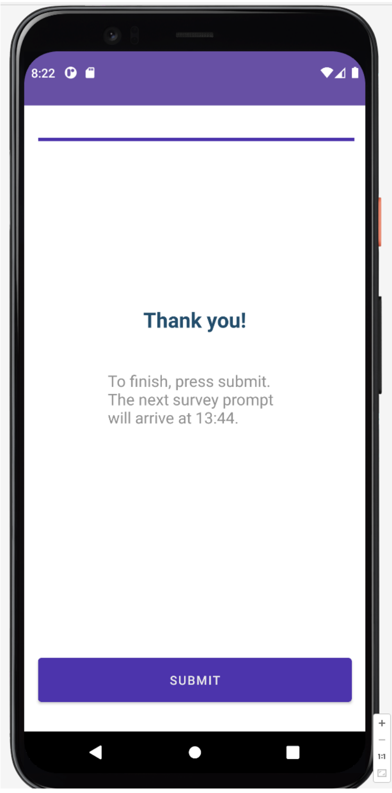

Figure 3.8: Fatigue Survey Notification Warning

The code that demonstrates our alarm scheduling can be found in the survey end activity in appendix B.1. In this scenario, we use:

```
alarmManager.setExactAndAllowWhileIdle(...);
```

We found this to be the best way to set as precise an alarm as possible. This will require the user to complete the survey when it arrives to send another notification as repeating alarm notifications were always imprecise and sent whenever the system was awake.

3.4.3 Notification Controls in Settings

Once we had the notification system sending regular notifications, we wanted a way to control when it would send them. For this, we implemented some sliders in the settings page and a database table to store the notification times. We established three different factors as to when to send the notification.

- Interval - how often a notification is sent. (1h, 1h30m, 2h, 2h30m, 3h)
- Awake Time - the times of day to send the notifications. (6 am to 12 pm at the extremes, sliders allow 1h adjustments.)
- Do Not Disturb - a block in the day to not send a notification. (At least 1h interval, sliders allow 1h adjustments.)

This leads the app to then calculate when the next notification is sent. This calculation is given in a function shown in the code of appendix B.1. To explain simply, the app works out the next available time to send the notification, the interval length of time from now. The interval time is only taken off the available time. For example, if a 2h interval is set, the survey is finished at 10 pm and the awake time is set to 8 am to 11 pm, the next notification should arrive at 9 am. There are ways we can make this more random by adding a random number (+-15 minutes) to the end of the notification send time or always sending trying to send at least 3 notifications in the day which would not take long to implement.

3.4.4 Emergency Notifications

The final part of our notification system was designing and implementing an emergency notification system. This idea outlined by our customer explored how survey notifications should be sent in extreme scenarios such as large amounts of noise or activity. Implementing this idea was quite difficult and required setting up the app as a foreground task so that it would run constantly. With how far we got with this feature, we have been unable to fully test the functionality in extreme scenarios when the app gets put to sleep by Android itself to see if it will work.

The version we achieved was one that monitors the background volume. This requires the use of the microphone to measure the highest decibel level in 1 seconds intervals. If the app detects a decibel level greater than a given threshold, it will set off a notification. The notification itself is on a roughly

20-second timeout to prevent being flooded with notifications. The code for the emergency notifications can be found in appendix B.1. We are using the MediaRecorder Android developer system [25] to set up the microphone. *MediaRecorder* is a state machine that will listen for a period where we can sample the maximum amplitude. Given that decibel is a dimensionless unit, we had to follow an arbitrary function to convert our seemingly null value amplitudes received from the microphone into arbitrary values that lined up with a linear scale. The following equation calculates our volume values, where the constant 32768 (the amplitude value limit) is used to bring the value up to a normal value.

$$Volume = 20\log_{10}(\frac{Amplitude}{32768}) + 60$$

This equation gave us values roughly in the range of 0 to 60, using 60 as the threshold for emergency notification.

The result here is not a fully complete emergency notification system but a proof of concept for definite. Ideally, we would like to expand this system to check the fitbit activity, number of steps, energy levels, etc. Adding these checks would be quite simple as most of the work for having this run in the background is now established.

3.5 Data Dashboard and Daily Summaries

As previously stated in the design section for the data dashboard, our final implementation would allow the user to switch between the independent 'daily summary' and 'lifetime' sections by using tabs located in a horizontal bar at the top of the page. This was achieved by adding an Android *Scrollbar* which would dynamically change the contents of the rest of the page depending on which tab in the *Scrollbar* the user has selected, by switching between two separate Android 'fragments' created for each of the two sections. In addition to using the *Scrollbar* for navigation, the user can swipe horizontally on each section to switch to the other.

Our implementation of the data dashboard as it appears in the application can be seen in the screen captures below.

Figure 3.9: Screen captures of the data dashboard, showing the summary page, the lifetime page and the lifetime page scrolled to the reaction time graph

Roughly the top third of each fragment is home to a *HorizontalScrollView*, which enables multiple objects to be placed in a horizontal row that a user can swipe left and right to scroll through each of the view's contents. This was chosen so that each graph individually could take up the full width of the screen when visible to avoid making the graph too small and hard to see or understand. The *HorizontalScrollView* contains three *LineGraph* objects created from the *Dude'sNamePackageOrLibrary MPAndroidChart* module to display each of our chosen data types. The first graph, [selected] to be first (and default viewed when you navigate to the page) as it is likely to be the most frequently viewed graph by the user, shows the energy level that the user responded to within the first question of the survey over time. This is accomplished by retrieving both the time that each of the user surveys in the database was taken for the x-axis and the user's response to question 1 for the y-axis. These values can then be plotted on the axes and joined by a line allowing the user to visually see the trend in their energy level over time. Similarly, the second graph fetches the result of each of the user's reaction time tests along with their respective dates and times. The same process is followed to retrieve and plot the FitBit data on its designated graph. The user is able to scroll sideways within each of the graphs to view data over the full timescale that they have been using the application, as the graph is limited to only display 4 data points at a time in order to achieve our goal of creating an uncluttered view of the data. Finally, each of the graphs was given a unique background colour to help the user uniquely identify them, with the energy level graph appearing in purple, the reaction time graph appearing in blue and the FitBit data graph appearing in green.

The remaining space on the screen below the graphs is home to the full information of how the user answered the questions in their past fatigue surveys. In front of a subtle off-white background colour, the user is shown a *VerticalScrollView* containing rectangular grey buttons for each of the user's previously completed surveys, sorted from top to bottom by the time the survey was taken. The button contains a small amount of text to display the date and time of the survey which crucially is able to uniquely identify each of the survey responses. The user can click the button to open a pop-up, an example of which is shown below:

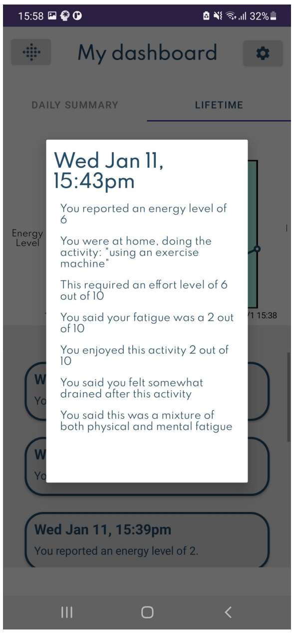

Figure 3.10: An example popup

The screen captures show how the full survey response data is fetched from the database and presented on screen to the user. Notably, the data is presented as a sequence of somewhat informal sentences rather than an unappealing list of simple keywords. Furthermore, each of the buttons contains the energy level from question 1 of the survey so that the user is able to see this data at a glance without having to click on the button to view the full pop-up. We liked this feature as it helps the user to mentally distinguish each of the reports without being presented with just a dull, data-intense list of date and time information in a list.

The layout is similar for both the 'daily summary' and 'lifetime' sections, with the crucial difference being that the data displayed on the daily summary page is only taken from surveys from the most recently completed day. This is defined by the most recent calendar date which has reached the time of day (specified by the user in the settings page) of the daily summary notification. For example, if the user has set their preferences to receive their daily summary at 10 pm and it is currently 9:31 pm on the 25th of December, the user would only be able to their survey responses from the 24th of December in the daily summary section. However, half an hour later at 10:01 pm, the user will have received a notification that their daily summary for the 25th is now available to view in the daily summary. Regardless, in both cases, the user would be able to review their recent survey responses from earlier in the day on the 25th in the 'lifetime' section without having to wait for the daily summary for the current day to become available.

The top left of the data dashboard contains a button to take the user to a page where the user can integrate their FitBit account with our application in order to view statistics on the dashboard graphs as previously seen. Additionally, the top right of the dashboard is also home to a small button with a cog which when pressed will navigate the user to the settings page.

3.6 FitBit Integration

3.6.1 Selection of Device

For the sensors/wearables part of this project title and requirements it was quickly decided that using an already on-the-market wearable would be preferable to designing our own wearable from scratch, this was for a few reasons:

1. We are only given a few weeks for this project so any wearable we design would likely be a very early prototype which is a problem because:
    (a) Our customer intends to use the product of this GDP for research so having something that is unfinished leaves them unable to do that
    (b) Even if the prototype was somehow fully functional and worked perfectly there would only be one of it which isn't helpful as even if they managed to run multiple participants with this one prototype was given each of the participants would need a few weeks or months with it it would take years to end up with a sample size still too low to publish any meaningful research from.
2. Massive companies like Garmin and FitBit will have access to huge labs full of R&D and will be able to produce sensors and wearables many orders of magnitude more sophisticated than anything we could make.
3. A lot of people already own smartwatches and other wearables such as FitBit which means much less money would have to be spent by any researcher using our GDP as a lot of the participants would already own the equipment needed.

Taking into account the decision the next design choice to be made is which one to focus on integrating with, this is because sadly all these companies use different proprietary software and format their data differently so you cannot create something to work with all of them at once and due again to the time constraints of this GDP resources need to be focused to where they will be most productive. This led us to decide to use FitBit. The reason for this decision is that two companies own over 80% of the smartwatch market - Apple with 46% and FitBit with 38% [30] which means if we want to pick a wearable a large amount of the population will have we will have to pick one of these two. Of these two FitBit is the better choice for a few main reasons. One of these is price, FitBit devices are considerably cheaper than Apple watches which can have prices up to £1,200 with even their cheapest device with the cheapest configuration still being £249.99 [21] Meanwhile even buying a brand new FitBit direct from their website at full price you can get them as cheap as £49.99 [24] which is important both for helping our customer get their research done for an acceptable price and for this GDP where we have a relatively small budget which if we wanted to buy one of these devices for testing, even the cheapest apple watch would eat our entire budget. The second main reason for picking FitBit over apple is that it contains a lot more of the info we need as it is a true Fitness wearable that contains info we need. This is opposed to Apple which is designed to be a more general-purpose smartwatch and whilst the higher-end watches that cost around £1,000 do contain most of the features the FitBit devices have the £250 one which is within our budget is missing key metrics such as being unable to give ECG (Electrocardiogram) data[20]. Even if this data was recorded by the device Apple makes their devices almost deliberately obtuse to work with any other device or program which isn't made by Apple to try to encourage consumers to stay within the "Apple Ecosystem". This could make it difficult for us to make our system work with theirs and get the info it needs.

3.6.2 Integration of Device with the App

Now that it was decided that FitBit devices would be used as the wearable portion of this project it was important to consider how this will be done. The original plan was to implement some sort of Bluetooth connectivity so that while you are using the app it could connect to the wearable and get data from it. This however turned out not to be possible as this is not allowed by FitBit due to security concerns. As this was not possible we decided to see how FitBit themselves do it within their dedicated app as they are likely to know the best way to do app-device connectivity with their wearables. On their app, the way they do it is through a "sync feature" where you log in using your FitBit credentials then press a "sync" button, shown below, and then the app pulls the information from the FitBit API.

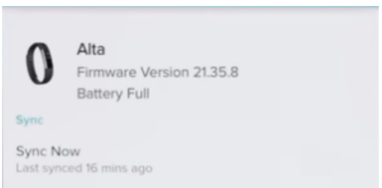

Figure 3.11: FitBit App Sync feature

FitBit gives, on their website, a tutorial on how to use their API for personal projects, apps and for server applications with different protocols for each. The process works as a slightly adjusted version of the common Oauth 2.0 protocol commonly used for many different types of APIs. The steps you need to complete to get the access token you can use to pull data from the API are as follows.

1. Redirect the user to the FitBit login page, this could be done through an embedded browser in the app however FitBit will eventually send the user back to a URL of your choosing so this means you can use the Android deep-link system to pull them from a browser outside your app. Therefore for the sake of security and to reduce the number of things that can break in your app it's best to direct them to the FitBit login page on the phone's default browser.

2. after this they will be taken to a consent page where they are shown a full list of all the data your app will have the ability to access and must manually approve each of the options. This again can be done either somewhere of your own choosing or on FitBit's official site, using FitBit's default one is usually better as in the FitBit terms of service there are dozens of rules you have to abide by such as all the boxes have to be unchecked by default when the user reaches the page.

3. after they have checked all the boxes and said yes they will be sent to your redirect link - in our case "app://www.brainfatigueapp.com/dashboard" a deep link that instead of taking them to a web page will pull us into an intent filter set up to catch it on the dashboard page of our app.

4. along with the redirect link the URL they are sent to you with will contain an access token only valid for 10 minutes as a parameter.

5. using this access token and your app's client id and secret concatenated and base 64 encoded you can send an HTTP request to FitBit's Oauth 2.0 server which will take your token that is valid for 10 minutes and in exchange send you back an access token that is valid for 8 hours.

6. After this you can start making requests to the FitBit API and as long as you have the access token in the header and it isn't expired yet you can access all the data for the user that they consented to on the consent page.

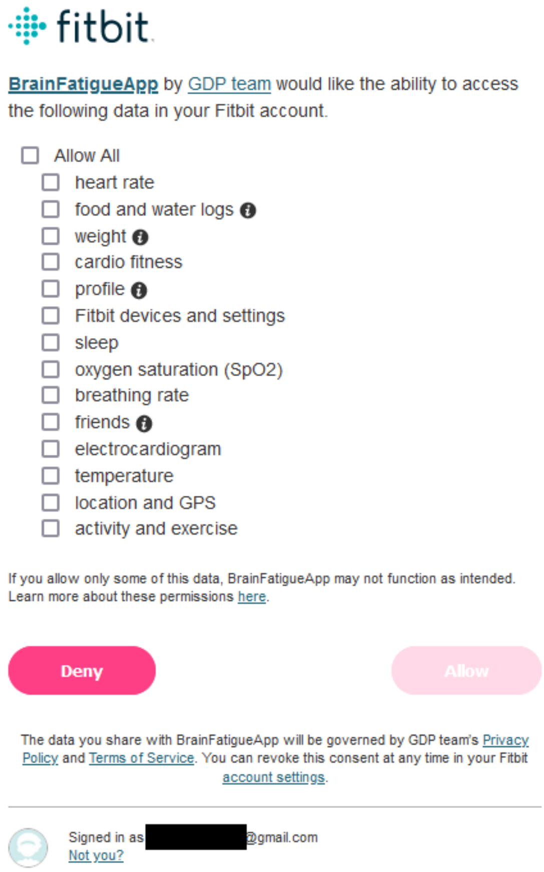

Figure 3.12: FitBit's User Consent page

The range of data you can gather from the API is quite vast however there are some limitations that we have. One of these is that for the intra-day data such as heart rate on a minute-by-minute level, you have to send your App off to FitBit along with a report of what purpose it has[22]. This report is quite extensive - expected to be around 32,000 characters[22] - so for this app we did not apply to FitBit for the extended range of information access and simply worked with what we could access normally which still includes things like food and drink intake, exercise minutes per day, number of minutes in each heart rate range per day and sleep. A full list can be found on the "FitBit web API explorer" [23].

3.6.3 FitBit Display on Dashboard

There were two spaces left on the dashboard for FitBit graphs, one in the "daily" section and one in the "lifetime" section so we needed two sets of data we could pull from FitBit that could be relevant to fatigue that works on each of these timescales. For the daily graph, the heart-rate for the day was selected. as was mentioned before we couldn't get the minute-by-minute raw data as doing that requires submitting a research request to FitBit and is a long and tough process normally only done by large companies. However, FitBit will still give you the data for the day in the form of how many of these minute-by-minute recordings fall in each of the 4 main heart rates ranges they record - "Resting/ Out of Range" (these two terms are used interchangeably by FitBit), "Fatburn, "Cardio" and "Peak". This was good as there have been studies showing that fatigue can cause changes in a person's heart rate[4] so it may be of some use to the researchers. If we had the full data we, of course, would be able to create a graph that would flow quite nicely and most likely give some sort of curved distribution however as we only had ranges the completed graph ended up looking as below.

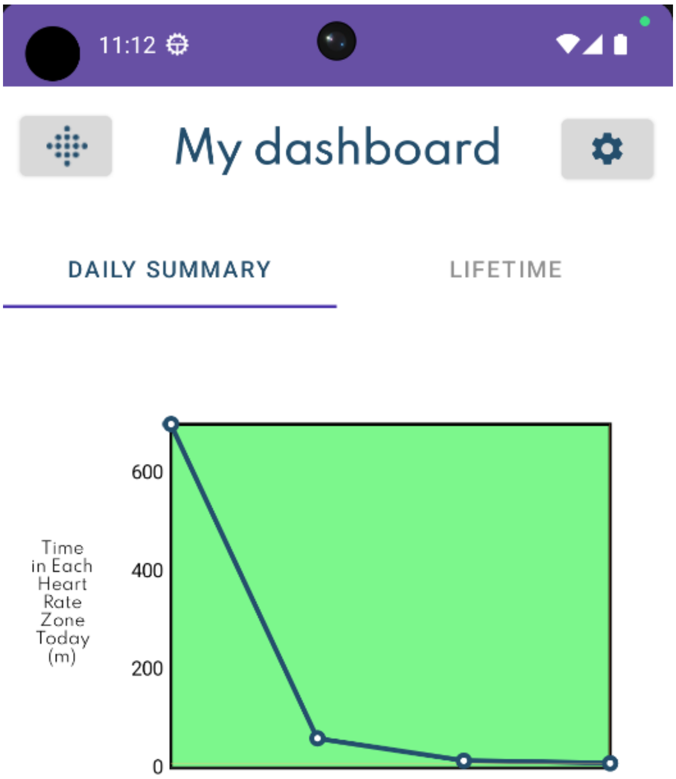

Figure 3.13: The FitBit daily graph displaying some example data

Which is not as nice as it could have looked however should still be enough to provide utility. The Graph for the lifetime page ended up looking similar however because small changes in heart rate in a day won't be visible on a lifetime scale so we instead decided to put the number of minutes of recorded exercise activity in each day. Having one data point per day means it shouldn't get too cluttered unlike the heart rate one which has multiple data points per day would do on this scale and it should still provide utility to the researchers as exercise also often correlates with mental fatigue [8] (not always negatively, for certain types of fatigue exercise can actually help greatly and is a recommended treatment[11]). The finished graph ended up looking as below.

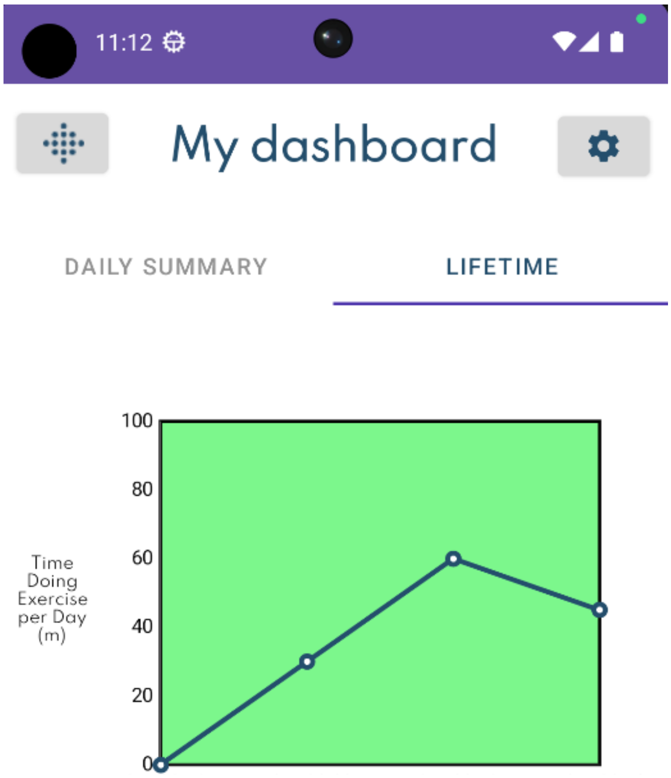

Figure 3.14: The FitBit lifetime graph displaying some example data

3.7 Storing the Data

With the amount of data we are collecting for the app, we needed a way to store it that wasn't going to be too costly to the performance of the app. Android offers a few ways to store information but the key technique we used here was to create a database using Room [27].

3.7.1 Room Database

The database design, as shown below, follows the schema recommended by Android Developers to follow. There is one abstract database that uses multiple Data Access Objects (DAOs) that use custom-written SQL statements to interact with the database. Within the database, there are a collection of tables to store several Java objects. In this case, we are storing the results of the survey, the settings page and the reaction results.

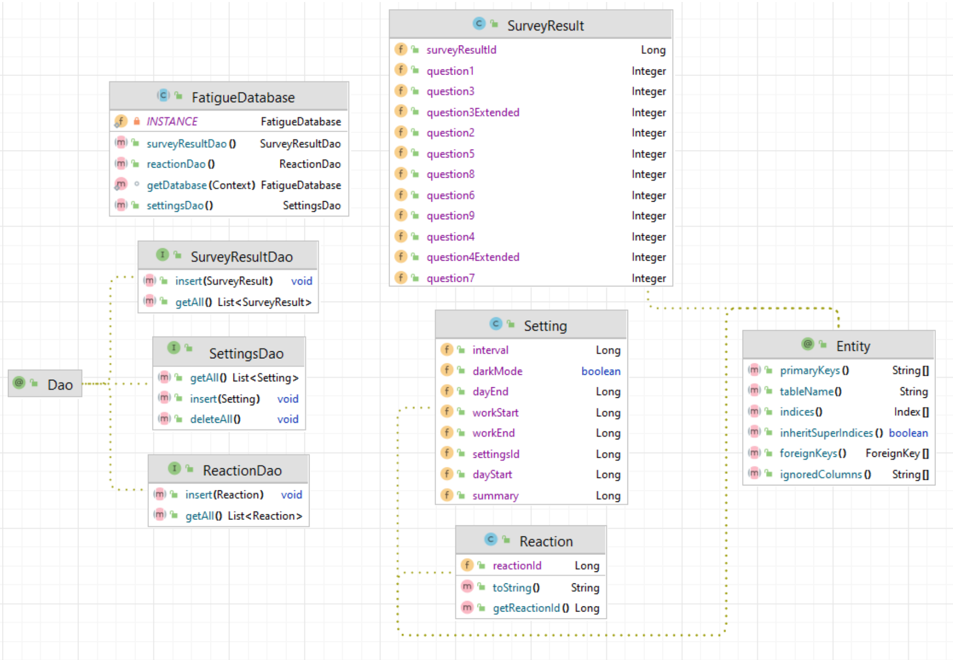

Figure 3.15: Room Database Tables

The database must be accessed on a separate thread than the app runs on, so we had to be careful in how we used it. For our scenario, we decided to use Java Executors which run in parallel to the rest of the app. For those scenarios when we wanted to wait for information from the server, we used Java Futures to receive a promised result.

Whilst most of this seems overly technical, the result is a very efficient way of storing the data. We noticed that we were easily able to upgrade the database over time using Room and we also noticed the data within would persist beyond just a few changes to the app's code making it resistant to any updates we made. If there was anything we could do to expand this feature of the app, it would be to start encrypting the data. Whilst an actor may have a hard time getting into the database, it'd provide an extra layer of security that could be quite easily developed. We are not storing a user's personal or sensitive data within this database so it wasn't a critical objective for us.

3.8 Login Page

The layout originally planned for the login page and implemented in the first few iterations of the app is shown below. It contains a username and password field along with integrated logins with google and Facebook. The reasoning for the integrated logins is that it has been shown in studies that people with traumatic brain injury can have impaired memory which means having a system which doesn't require them to remember another login could be immensely helpful[2][7]. The reason Facebook and Google were chosen specifically is that a Google account is guaranteed to be at hand for all users of our app as it is the account type Android uses and Facebook is the most popular social media amongst the older portion of the population who are those more likely to suffer from traumatic brain injury[16][18].

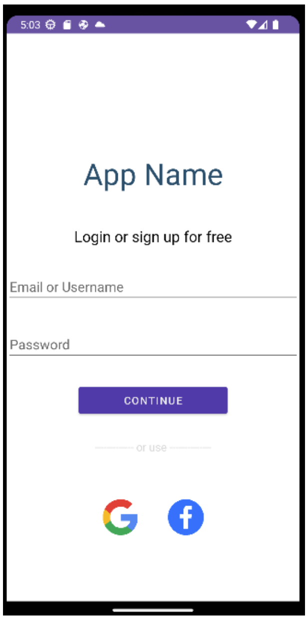

Figure 3.16: Room Database Tables

The first major update and deviation from the plan happened fairly early on in development when we realised if we set up a system for the username and passwords to be stored somewhere securely it would probably require some sort of upkeep cost and/or supervision from a developer. This would not suit our customer's needs as they intend to use this for research after this GDP is over and the team who worked on this app have moved on to other things. After consideration of this the username and password fields were removed and in their place was added an option to continue to the app without logging in. Selecting the continue without logging in the option means fewer data can be accessed and some personalisation features will no longer work, for instance, they will see a generic greeting on the homepage rather than a personalised one, however, these are cosmetic and do not affect the functionality of the app.

The second update to the login page after its inception came during testing in which it was discovered the Facebook integrated login system we used was deprecated by Facebook themselves as an anti-phishing measure. This meant it had lost all functionality unless additional steps were taken by the Facebook user to go into their account settings to re-enable it. This is discussed further in section 4.1.1.1. After this was removed we were left with just two options- google integrated login and continue without login. The small icon for the google integrated login that can be seen in the first iteration was also decided to be too small so these were changed into thick bars that go almost the full width of the screen which should help those with issues with their visual and motor functions to use it more easily. This was the final update to the login page which left us with the page as it looks today which can be seen below.

3.9 Settings Page

The creation of the settings page begun with the addition of the drop-down menu and three sliders that make up the controls for the user's notification preferences. The notification frequency drop-down menu was located at the top of the page as we felt it was the most important of the settings, and the user can simply tap on the menu to bring up the full list of time intervals between notifications and then tap again to select their preference. Below the setting for notification frequency, we implemented two Android /RangeSlider/s that allow for the adjustment of two movable thumbs on a movable track, allowing the user to define the lower and upper bound of a given time range.

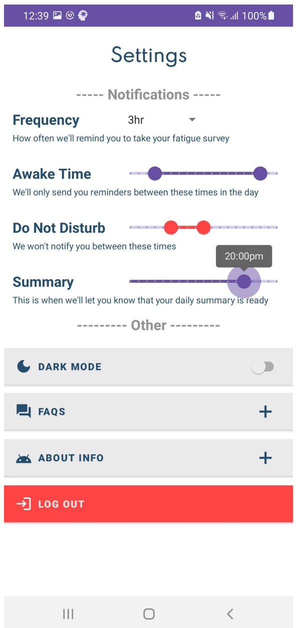

Figure 3.17: Settings Page

It was especially important that the three sliders were vertically aligned to each other, as it would make the user's experience with them a little more intuitive. This is because each of the sliders has the same lower and upper bound of 5 am-12 am, as the user sets their preferences they can see an accurate visual representation of their selected time periods in the distance between the thumbs of the sliders.

The settings page is also able to communicate with the database. We created a Settings table in the database that stores records of past settings configurations. This is used to save the user's previously entered settings so that the app is able to load succinctly and with the same settings as before. Every time the slider is changed or the dark mode is switched, a new settings entry is sent to be entered into the database. The settings page updates itself by taking the most recent entry in the database. In order to reduce the storage used by this system, the database can delete old entries in order to not waste space.

The first button in the lower grouping of settings options toggles the application's dark mode on and off. When pressed, the switch on the dark mode button inverts and the application switches the theme to change the primary colours to different options and changes the background to black, a before and after comparison from the application is shown below. We discovered through testing that this is not as rigorous as we had hoped it to be, as the dashboard does not display dark mode very well due to the colours contrasting quite badly. However, updating this theme in the future would be fairly simple as it only requires changing a few hex values to make a design that works well in the dark.

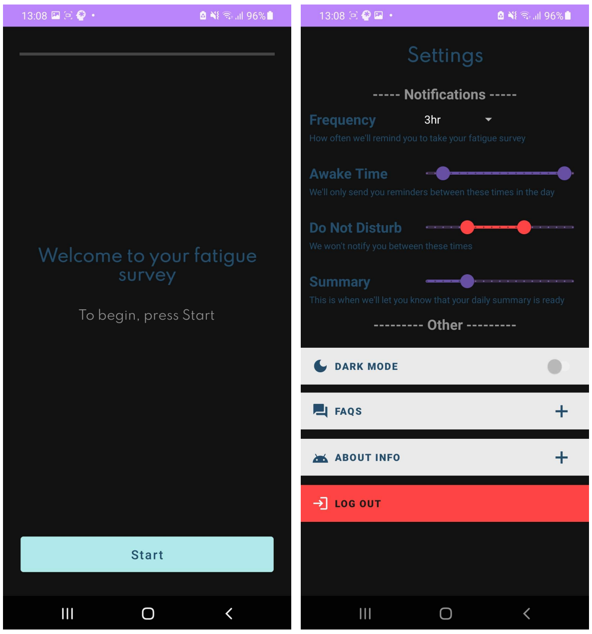

Figure 3.18: Dark Mode Examples

The next two buttons in the lower group on the settings page are the 'About info' and 'FAQ' buttons. Similar to the pop-up for displaying the full data of survey responses in the data dashboard, pressing either of these buttons will create a pop-up containing the application's 'About' information or Frequently Asked Questions. As before, upon creation the screen behind the popup fades and is dismissed when the screen is tapped again.

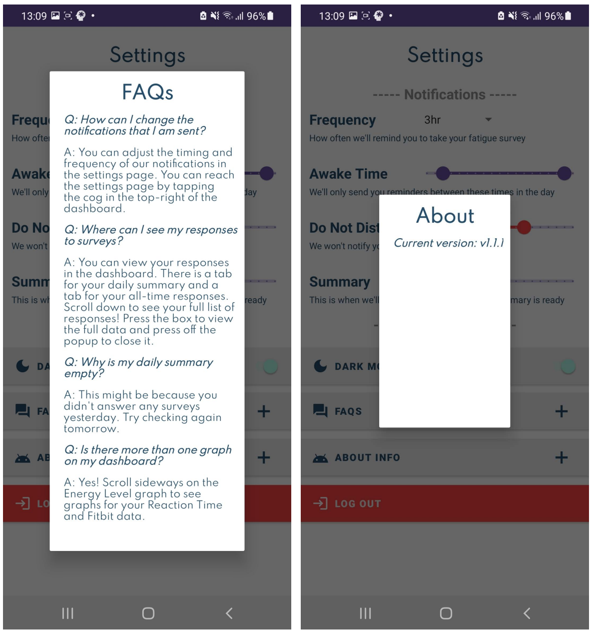

Figure 3.19: FAQ and About

The final button on the page allows the user to log out of the account that they are signed in to. We included this button as it is important to allow the user to opt out of having their account linked to the application when or if they want to. When the user presses the log-out button, the app revoked access to the user's account data and the user is returned to the login page at the beginning of the application.

Chapter 4

Testing

4.1 Internal Testing

For the internal testing of an App of this nature where multiple people are contributing their own sections, it is important to have tests that both cover each person's individual contributions along with making sure that each person's sections work together without issues arising. To cover these main areas the testing was split into the commonly used Unit, Integration and System testing system which splits the testing into these 3 sections:

Unit testing, which is the testing of small components of the program, usually made by just one person, in isolation. These tests are useful as they can often be made simple and automated so that they can be run every time a change is made to make sure nothing has been broken during the changes.[1]

Integration testing, which starts running the individual units together in bigger sections to try to uncover any issues in their interactions which cannot be seen in unit testing alone. This is especially important in an environment where multiple people are working on different units as they may have had different ideas of how the units will work together which can cause unintended behaviour in the app.[1]

Systems testing, which is the testing of the full, completed system and making sure that it meets its requirements. In our case, this means testing and using the app in its entirety, in an emulator installed on a mobile to make sure it provides the functionality required for our customer [1]

4.1.1 Unit Testing

A non-exhaustive list of some of the unit tests carried out is as follows:

Test name	Description	Expected outcome	Result
Google Sign-in button	Creating the Google sign-in button which will send you to the google integrated sign in	Screen should become greyed out and a menu should pop up listing all your Google accounts awaiting you to sign into one of them	Pass
Facebook Sign-in button	Creating the Facebook sign-in button which will send you to the Facebook integrated sign in	Should navigate you to the inbuilt browser to log into Facebook	Fail - Discussed below table

4.1.1.1 Facebook Sign-in Problems

Creation of the integrated Facebook log-in for the app went smoothly as the Facebook documentation explains how to do it in a simple and straightforward manner. However, once we began testing it the login simply would not work, upon researching why this is it was discovered that in October of 2021, Facebook decided to disable sign-in from embedded browsers due to a sharp increase in phishing attacks using the feature.[32]. This can be fixed by making the person log in go into their user settings on Facebook and re-enable it however this is something not all users of our app would be able to do necessary and even if they did manage it this would allow them to be targeted by the phishing attacks Facebook disabled this feature in order to prevent. There are other ways to log in using Facebook for instance if they have the Facebook app also on the device you can use the "Android App Switch" feature to log in. However, as this would likely raise new issues, such as what happens if someone doesn't have the app, and the fact that everyone on an Android phone is guaranteed to have a google account we decided to instead remove the Facebook login from the Sign-in page.

4.1.2 Integration Testing

Test name	Description	Expected outcome	Result
Going Through the survey	Going through the survey so make sure each individual question navigates to the next one correctly	Should go one by one through the survey in a way that follows the layout designed in Figma	Pass
Checking the dashboard screen for past surveys	Check the dashboard screen to see that surveys you have taken in the past are displaying there	If the database has integrated with the dashboard correctly you should be able to see all your past survey results	Pass
Notification upon results being ready to view	Checking that once the results of that day's surveys are ready to view for that user they are sent a notification for it on their phone.	A notification telling them about the results being ready should pop up at the top of the screen	Pass
Test redirect away from and back to app works	Check that the redirect that takes you to the FitBit login page passes FitBit the correct URL to redirect you back to the app after login	if the link FitBit redirects you to after login matches a deep link associated with the app then it should bring you straight back into the app	Fail - Discussed below

4.1.2.1 Android Deeplink issues

Android has tutorials for creating a deep link in their documentation so these were followed and everything seemed to be going smoothly, the path "https://www.brainfatigueapp.com/dashboard" was associated with the app's dashboard so that if you had the link in say, your phone notes app, had our app installed and clicked on it, Android would give a popup saying "would you like to open this link with brainfatigueapp" and once clicked would navigate you to the dashboard. As this worked fine it passed the unit tests for itself. However once we integrated it with the FitBit login it suddenly stopped working, instead of opening with our app it would just search the web for "https://www.brainfatigueapp.com/dashboard" which is an unregistered web address so it would throw an error. This confused us for a while because we checked FitBit was using the correct redirect link and it was so there was no reason it couldn't open it with our app. The answer came when looking through the documentation for how Android chooses how to open an app, up until now we had been relying on the fact that when clicking a link Android will go "which app do you want to open this with" however it turns out there is an exception to this rule if you are already in an app that recognises that link format it will just open it with that app. This makes sense if you think about it as you don't want to type a link into chrome and then get asked "would you like to open this with Firefox". The solution to this was to change the deep link to a host the browser the fitbit login was happening in wouldn't recognise so it forced Android to not use the currently open app and search which apps could open it. So now instead of "https://www.brainfatigueapp.com/dashbaord" it would redirect to "app://www.brainfatigueapp.com/dashboard. This has the unintended bonus of our app now being the only app on a phone that will recognise the format so now there isn't even a popup asking "which app would you like to open this with", it will just immediately take you to our app.

4.1.3 System Testing

Test name	Description	Expected outcome	Result
Complete a survey to show in the dashboard	Go all the way through the survey give answers and then navigate to the dashboard to see the results	If this works as intended a new instance should be added to the list of past surveys and should reflect the answers you gave	Pass
Logout and Login	Use the logout button in the settings menu to logout and then once you are on the login screen login with google again	If the logout worked correctly when you press the login with google button on the login page instead of it remember you and immediately navigating to the home page it should ask you to enter your details as if it were your first time logging in.	Pass
Test the notification time frames	Go into the settings and adjust the time frames for when the app is allowed to send you notifications.	The app should send you your next notification after the no notifications time you specified is over	Pass.

4.2 External Testing

4.2.1 Ethics Approval

One of the aims that were set out as part of the project scope was to ask the public to test the end product's functionality and usability. Before submission of the ethics application form, there were a few key factors that needed to be considered: the right number of people, what group of people we were planning to ask, how many days the person would be using the app, the types of data being displayed on the app, how we were planning on obtaining the feedback and steps on how users will be using the app.

As the testing and gathering of feedback on the application could only be done once the development of the app was completed, we decided as a team that 5-10 people would be just the right amount to document how people felt about using the app and the improvements that could be done if there were any. After deciding the number of people, we then had to decide if we were going to be asking students from a specific department such as the Occupational Therapists (OT) department, or just generally around the university campus. The OT department was considered as an option because according to studies and research done by our customer, the OT department was very much involved in the process of developing the predecessor app as they had a better understanding of patients that had sustained brain injury before.

The other option was to ask patients that have had acquired brain injury before. However, our customer and supervisor told us we wouldn't be reaching out to patients for this project and we were not required to assess the answers logged by the users in a clinical manner. We were essentially developing a tool to aid our customer in doing so. Therefore, user feedback only focused on gathering different opinions on ways in which the app could improve, thus we decided to limit the scope of testing the application to general students around the university or with students from the operational therapist department. The proposed idea for testing was to have the app installed on their personal phones or have it installed on a separate Android phone which would mainly be used for testing. Then, have users also link the FitBit to the app for over a period of five days to test whether mainly: the notification alerts would prompt the user at different times of the day during the testing period and if the data from the FitBit was synchronised and displayed on the app.

Before the testing period commences, participants will be asked if they would like to partake in this study and if so, they will be given a Participation Information Sheet (PIS) which contains the outline of what the study is about and why the study is being done. This allows participants to gain a better idea of how the study will be conducted before deciding to participate. If the participants are willing to participate, they will be given a consent form and it is mandatory for the user to sign it before the testing period commences. Then, the participants will be taught how to navigate and operate the app on a mobile phone. The testing period will be five days, during which the user will be prompted by notification alerts throughout each day to answer the activity survey questions:

- Question One will ask the participant how they are feeling in terms of energy levels on a scale of 0-10.
- Question Two will ask where the participant is at the moment.
- Question Three will ask what the participant is doing right now. The participant must pick the closest category.
- Question Four branches into more specific activities based on the activity chosen in Question Three. (these specific activities can be seen in appendix B.3.)
- Question Five will ask how effortful the activity was in Question Four take.

- Question Six will ask the participant how fatigued they are at the moment.
- Question Seven will ask the participant how much they enjoyed the activity.
- Question Eight will ask the participant how the activity affected their energy levels.
- Question Nine will ask the participant to choose the option that best describes their fatigue from physical, mental or a mixture of both fatigues.
- Question Ten will ask the participant to take an interactive reaction time test.

Throughout the testing period, the participant will be able to view a summary of how their fatigue was affected by the activities they did throughout the day. This will be presented to the participant as a separate dashboard section of the app. The steps are then repeated throughout the testing period and once it ends, the participants will be asked to take part in a survey which allows them to provide any type of feedback regarding how they felt about the app's design, the activity questions being asked, the overall transition of the app and anything, in general, can be voiced through this survey.

Since a major part of the project involved incorporating wearables and sensors such as a FitBit, we had to consider how long each user could test with the FitBit as we had a limited supply. One recurring issue with this was the time constraint that we were dealing with as it was quite impossible to have all participants take turns using the FitBit, the testing period would last for more than a week in total. Hence, user feedback focused on collecting users' opinions on how user-friendly the app was if any part of answering the survey questions caused stress/confusion and additional comments if they had any.

There were a few points that made us reconsider submitting an Ethics Application for the testing of the app. This was mainly because of the app potentially not being fully complete, and the possibility of our ethics application being rejected. If the app was not fully done, we felt like there wasn't much point in having to test the app. If the ethics application kept coming back disapproved, there was going to be more time spent on revising the application and resubmitting it again when this time could have been used on developing the app instead. Even though the testing of the app was the final leg of the project, there was still a possibility that the ethics would be approved and the app being completed thus, we decided to submit it and work on the app simultaneously.

4.2.2 Setbacks

The first submission was disapproved as parts of the application made the study misleading namely, the study title which was the title of this project, made it seem like participants will have experienced brain injury and were not reflecting what the study was intended to do. There was also a lack of detail explaining how participants will be wearing sensors/wearables when they take part. Along with the main ethics application form, there was a document called the Participation Information Sheet which lacked detail in which, just stating that the participants were volunteers was insufficient. The ethics committee pointed out they needed to know more information about inclusion/exclusion criteria i.e, age, and the link to brain injury which again was a confusion led by the title of the study. There were other smaller adjustments that needed to be made such as the formatting of the PIS as there was a mixture of font colours and deletable information that was not deleted. Lastly, the risk section was supposed to be in terms of the participants, but it was written in terms of risk with the technology, however, it was suggested that it may link to risks with the participants like frustration.

The ethics application was edited based on the feedback and resubmitted with the title changed to: "Evaluating the usability of a mobile application designed to help people track their levels of fatigue". This came back disapproved as well with similar feedback mentioning that the study design within the application still implies that brain injury patients will be the participants. It was advised that our sentences had to be edited in a way which mentions that the app will be developed with a non-clinical population and had applications in the treatment of brain injuries. The ethics committee also wanted the 5-10 participants from the Occupational Therapy Dept. to be made clear whether they were to be staff/students and not patients. Another area that needed more refining was the section mentioned previously in the PIS, asking for the characteristics of the participants as stating they were volunteers was insufficient. The ethics committee found that age and additional inclusion/exclusion criteria such as gender and visual function alluded. Thus, it had to be made clear that gender was not an eligibility requirement at all in the PIS and participants who felt like they did not fit the criteria stated in the consent form they could choose to opt-out. After editing the ethics application form according to the feedback, the third attempt came back successful.

In summary, the ethics application was successful after the third attempt at submission. This was mainly due to the constant implication of the study requiring participants to have had a brain injury before in order to be eligible to participate and not being specific enough about the characteristics of participants i.e inclusion/exclusion criteria. The issue was then solved once the title was changed as this removed any implication towards having users that have had a brain injury before being the only eligible participants. The other issue which resolved the submission as a success was further clarifying the characteristics of the participants as well as mentioning that gender was not considered as an eligibility requirement.

4.2.3 Feedback

In order to use the ethics application, we create a short quiz that we used to test the app with a few friends and family. Whilst this is significantly less beneficial than what we had planned with testing with the occupational therapy department, we had to make do with the time we had as a result of the ethics application. We designed a six-section Google Forms quiz with each section comprising of the same questions about different key pages of the app. We showed them images for the following pages, as well as in some cases getting them to use the app if we had it downloaded on a test phone:

- Homepage
- Battery Survey Page
- Multiple Choice Survey Page
- Reaction Time Tests
- Dashboard
- Settings Page

For each of these pages, we asked the following questions:

- Do you feel that this page/screen from the app is user-friendly?
  - Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree
- How confused are you by what this page/screen from the app is showing you?
  - 0 to 10
- How stressful do you feel this page/screen from the app is to use?
  - 0 to 10

After every section of questions, we provided a space for people to leave feedback, questions and suggestions. This was most beneficial to us to make a few last improvements, mostly to the user interface and user experience.

4.2.3.1 Results

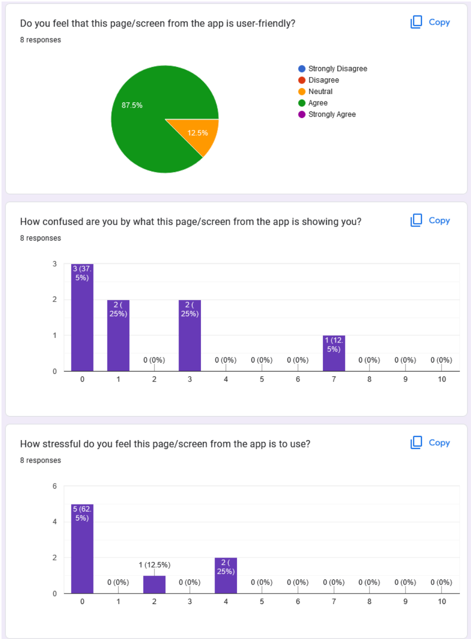

Figure 4.1: Section 1

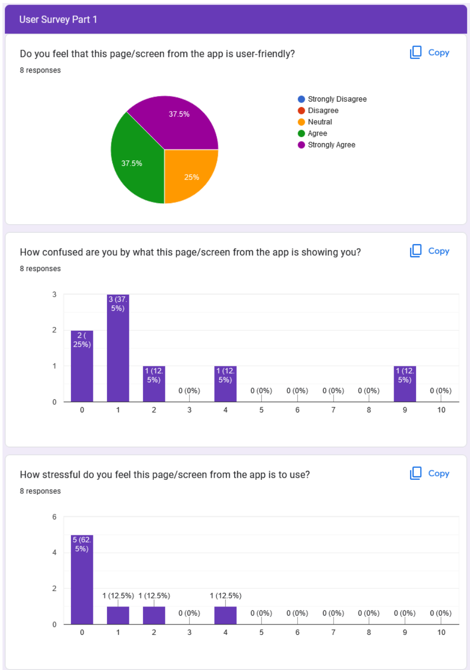

Figure 4.2: Section 2

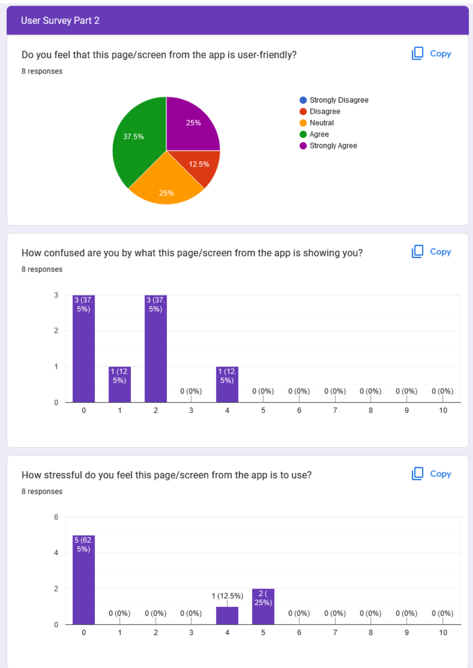

Figure 4.3: Section 3

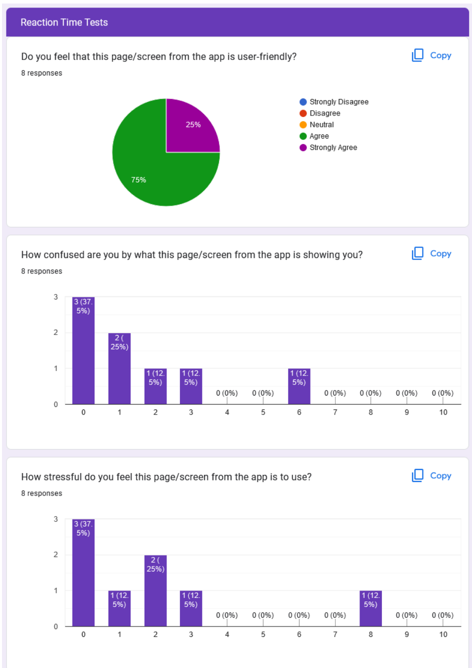

Figure 4.4: Section 4

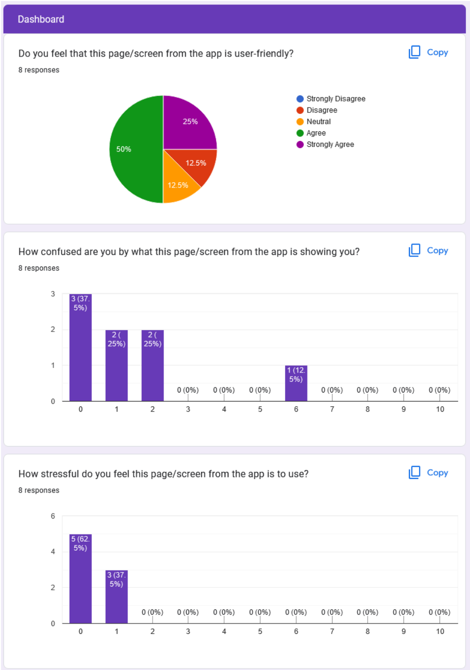

Figure 4.5: Section 5

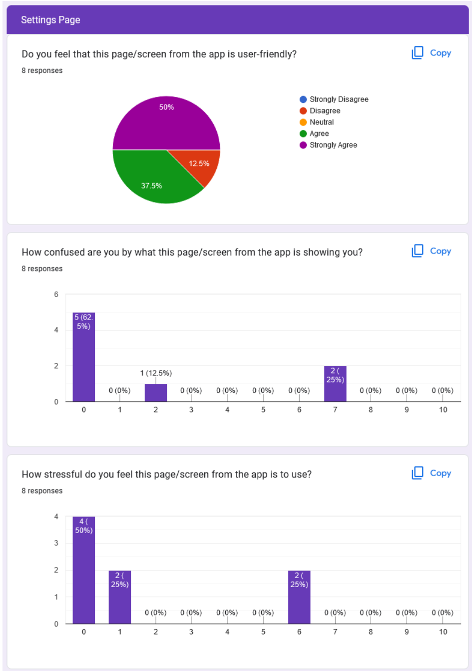

Figure 4.6: Section 6

4.2.3.2 Analysis

From these figures we can take the following metrics:

- User Friendliness
  - Strongly Agree - 27%
  - Agree - 54%
  - Neutral - 12%
  - Disagree - 6%
  - Strongly Disagree - 0%
- Confused
  - Average of 1.77
  - Participants were mostly confused about the battery slider question due to the static image; least confused about the user survey, PVT, and dashboard.
- Stress levels
  - Average of 1.04
  - Dashboard is the least stress-inducing; PVT is the highest stress-inducing overall.

The most stand-out thing about this is that our app is slightly more confusing than it is stress-inducing, but we still have fairly low numbers here.

This average could be improved through a wider sample size but as only abstract measures, we are quite happy with these results. Our user friendliness was very strong on the whole, with only a few people finding it not user-friendly with no one totally disagreeing. Whilst these stats are useful, we can dig deeper into the additional comments.

The most common bit of feedback we received was about the user survey battery. People were interested in the battery reflected in the slider value and this was certainly something we would have liked to implement. We also received a couple of comments about the battery being on a scale of 0 to 100 which felt quite confusing so we later made a change to lower it from 0 to 10. From this feedback, we decided that the resolution of 0 to 100 was too much for an already subjective measure. Most of the rest of the feedback was on the UI design issues that arose, mostly from trying to translate our UI design to a mobile device with sizing and positioning difficulties. We acted on this feedback with some global colour scheme changes to provide a better theme to the app and redesigned a few features to appear slightly better. We got the best feedback in general for our reaction time tests which we still managed to make a few improvements to, hopefully making this a very strong part of our app. We received some feedback about the homepage which we, unfortunately, have not had the time to change as this design was developed very early on and never fully touched again. This could be improved in future by providing a clearer indication with the homepage and providing an extra link to the reaction time tests.

Chapter 5

Project Specification Outcomes

5.1 Technical Objectives

Before we started the design or development of any part of our project it was important that we clearly defined a list of objectives so that we would be able to monitor our progress and success throughout. We first laid out and ordered our list of technical objectives, which contains the core features of the application that had their development prioritised due to their presence in the final product being key to its overall functionality or due to their completion being required for another objective's development to begin. Following communication with our customer, our technical objectives were:

1. Design and implement individual daily summaries of user-surveyed data.
2. Design and implement a reaction time test(s).
3. Implement a system for automated data collection.
   (a) Include data from external devices such as FitBits or Apple Watches.
   (b) Include additional data metrics - environmental, phone usage, emotions, etc.
4. Design and implement user customisation systems for the app's functionality, appearance, and feedback.
5. Optimise parts of the app UI and UX to be as user-friendly as possible.
6. Test the prototype on the Occupational Therapy department/students.

Our first technical objective has been fully implemented. As has been shown across multiple previous sections on our implementation, the application is able to store the user's responses to fatigue surveys and later present this information to them in the form of a daily summary inside the data dashboard. Furthermore, the user will receive a notification that their new daily summary is available to view in the dashboard at a time that they can specify in the settings menu.

Our second objective has also been fully implemented with the inclusion of the reaction time test that is presented to the user following the submission of responses to a fatigue survey. The test will calculate the user's average reaction time over two minutes and then store the result so that it can be later reviewed in the dashboard, as detailed in previous sections of this report.

Our third objective has been partially met, with the integration of a user's FitBit account data into the dashboard. Development of this kind of data collection proved to be time-consuming and so we ultimately decided to complete just one of the sub-objectives which came to be the incorporation of FitBit data into our application. [Cite Max thing about FitBits being most common?].

Our implementation of a settings page has resulted in our fourth technical objective being fully met, where the user is able to customize both their notification preferences (including alert frequency and do-not-disturb times) and appearance (by toggling dark mode on and off).

We feel that our fifth technical objective has also been achieved for a series of reasons. Firstly, the colour scheme of the application follows a consistent blue-purple theme that is welcoming and not harsh to look at. Secondly, the addition of a small progress bar to the top of the fatigue survey helps to guide the user through the sequence of pages and prevents confusion when navigating back to change their answer to a previous question. Finally, our choice of font and wording of instructions has been chosen to maximize the application's clarity and ease of use.

In contrast to the rest of our technical objectives, our sixth and final technical objective was not met. For the reasons fully described in section 4.2 'External Testing', we did not carry out a study involving participants from the Occupation Therapy department and a working version of our application. We had however created a contingency plan in case this occurred and so we instead collected feedback on several screen captures of the application as is described fully, along with analysis and feedback, in the 'External Testing' section.

5.2 Stretch Objectives

We had 5 stretch objectives for this GDP after speaking to the customer at the start. These are objectives that don't have any effect on the customer's ability to conduct the research they need without the app but would make the user experience a lot more seamless. These objectives were:

1. To design and implement a long-term calendar tracking past results
2. To design and implement charts to show long-term fatigue progress and trends over varying timescales.

3. Implement additional compatibility for unusual/unreleased external devices.

4. Design and implement a new survey for significant impromptu symptom/activity recordings.

5. Design and implement an Android widget(s) to display daily summaries without opening the app.

The first objective, "To design and implement a long-term calendar tracking past results" was not achieved. The app does store the user's results long term and they can be accessed however they are shown in a list on the dashboard and there was no additional page for a nice calendar view of them implemented due to time constraints. The impact of this on the research should be minimal as the information is still there, just not presented as nicely as it would be if the calendar had been implemented. The main negative impact of this falls upon the user as they will not be able to see the long-term effects of certain activities on their fatigue. This should not have too much of a detrimental impact on this app however and in some cases may be a good thing. As is noted by B.M Hood (2010) and M. Shermer (2011) people have the predisposition to draw causation from correlation and as is said by Dobelli (2013) this predisposition "leads us astray practically every day"[9][14][15]. This could be problematic for the users of this app if they see a correlation between a particular activity and fatigue using our app's long-term trends and start altering their behaviour and avoiding things in day-to-day life which may not actually be the cause of their fatigue. Therefore, in retrospect, it is possible not having this feature is for the best as it means the only people who are likely to see the final correlations are the researchers who will be knowledgeable enough on the area to avoid making any false assertions.

The second objective is similar to the first but in a chart form rather than a calendar. This one was partially implemented as the app's dashboard does show a chart of your past results however there is no option to change the timescale it shows currently and because of reasons mentioned previously, it does not show the user any trends or correlations.

The third objective, similar to the last was part implemented. We were unable to implement compatibility for more than just the FitBit family of devices within time as every company has a different system so this would require completely rewriting a new copy of all the code for the FitBit integration but for these new companies' protocols and formatting. However the section of this objective about compatibility for "unreleased external devices." was implemented. This is due to the procedure mentioned in 3.7.2 where instead of using a direct connection to the device itself we instead have connected to the FitBit API and get the device's data from there. This means that even if new devices are released by FitBit, as long as they have their data uploaded to the API, our app will still be able to work with it.

The fourth objective Was not completed. The main app survey was the one given to us by the customer and is the one all the results on the dashboard are based upon. Whilst the idea of having multiple surveys to keep users engaged was an appealing one at the start, due to the fact the entire app revolves around this one survey and the whole purpose of the app is to gather data using this one survey, adding another survey that people could choose to take instead quickly became unfeasible and would require extensive consultation with the customer to make sure it still collected the core data they needed.

The fifth and final objective was not achieved. Android is incredibly restrictive with stuff you are allowed to show to the user while your app is closed. Even getting the notifications done was a challenge as Android likes to send the notifications out in batches and then leave the user alone before the next batch to try to prevent users from being spammed by an endless stream of notifications at all hours. Whilst understandable from a user experience point of view it can make getting notifications to go off consistently at a certain time a challenge, especially if these times are too close together as there are limits. Widgets are, understandably, even more, restricted than notifications because they become a semi-permanent fixture of your home screen. After researching and realising this, it was decided the time was better spent elsewhere. To compensate for this loss we have tried to make it very quick to access the dashboard from within the app. This makes the widget unnecessary as the widget would purely be duplicating the information on the dashboard if you can make it so that information can be accessed in around 5 seconds even without it then there would be no significant increase in time between the two options. As is shown in previous chapters the flow of the app is set out so that the dashboard button is one of the two main buttons on the home screen you see immediately as you open the app so access to it is quick and easy.

5.3 Gantt Chart Comparison

A lot has changed since our initial Gantt chart plan. Whilst we believe we met a large number of our technical objectives, our stretch objectives were less accomplished, although we have very much achieved a couple of them as they slowly formed part of our main goals. Our initial plan tracked a steady line of progress across the term, taking at most three weeks for each task. However, we discovered in practice that most of our development would become backloaded. This was not intentional but we could have planned better for this. We severely underestimated the dependency created by setting up the app in a medium none of us were familiar with. Even though we are comfortable with Java, getting used to merging XML designs and activity flow was a challenge. Shown below is the comparison of our original Gantt chart to the final plan. This data is based on our Git commit history and record logs of work done.

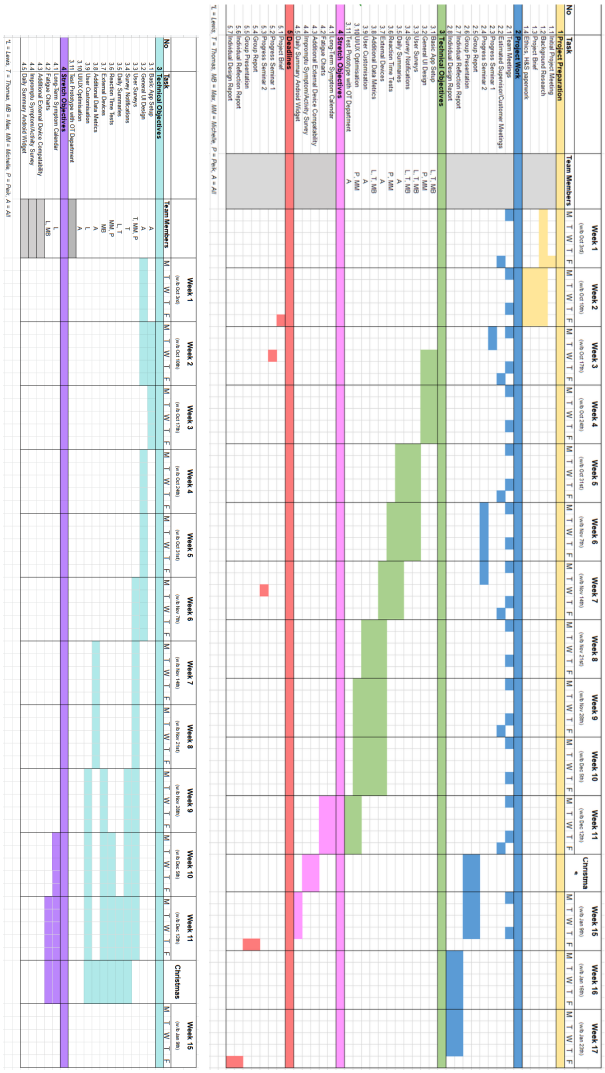

Figure 5.1: Gantt Chart Comparison

We can see from this graph that the bulk of the time was taken up by the user surveys and initial UI design. The user surveys were a pivotal part of the app so we spent longer on them to perfect them but also took our time in polishing so the hours spent later on were less and sporadic. We also discovered that because the user surveys provided most of the data, we needed to work on this task in conjunction with another. The best example of this was setting up the database to store information that could be collected by the dashboard. The UI design also has a lot of time logged because this task evolved into multiple others such as the login page, dashboard design and settings page. The time logged under other tasks also became these pages even if not specifically planned under that name.

Another big part of the Gantt chart we can see here is the time spent on stretch objectives. We discovered in early December that having a dashboard that displayed information was progressing well and began to quickly and efficiently expand this feature to provide graphs, long-term results and daily summaries. This started to come under our stretch objectives somewhat unintentionally. There are perhaps lots more things we could have planned for these stretch objectives goals but these could very easily be added to the dashboard.

Chapter 6

Project Management

6.1 Project Planning

6.1.1 Customer Planning

In the beginning, we tried to come up with a basic layout and idea of the parts needed for our project. A key stage in this process was organising a couple of meetings with our customer to understand the progress already made and where we could take those ideas further. Through this process, we set up a Teams channel for communication and meetings where we were able to receive a few documents from our customers, including their thesis as well as one specifying the user survey contents. We found this especially useful to establish a first draft UI design and begin setting up the app following a basic structure. A later meeting helped provide critical feedback on this UI design which we then expanded on and prepared to develop fully. Our communication with our customer was quite front-loaded as once we received a direction it was quite clear where we wanted to take it, but we occasionally conversed on some expectations.

6.1.2 Group Planning

Our group planning was established from our first in-person meeting. We set up a Discord group to provide an informal place to call, share files and discuss ideas. This platform became an essential part of our project planning and we all used this incredibly frequently. We discovered during the first month or two that in-person meetings were going to be tricky with one person having Covid for a week and another person spraining their ankle making getting onto the University campus difficult. During this time we realised our reliance on Discord. Meetings became much more regular and more impromptu.

6.1.3 Code Planning

As discussed during the development setup section, we used a private GitHub repository to all work collaboratively. To make sure we didn't conflict with our work, we set up meetings to organise who was working on what and created our branches to work on. When we finished that feature, we would pull, squash and merge that branch into the master whilst everyone else updated their branches with the new changes. This meant we could work collaboratively but also separately without breaking other people's builds. The usage of branches meant we could be very aware of everyone else's progress and work planned.

6.2 Division of Responsibilities

6.2.1 Technical Contribution

Contributor	Contributions
Lewis	- Data Dashboard Design
”	- Dashboard Energy & Reaction Time Graph Development
”	- Dashboard Daily Summaries and Survey Review Areas
”	- Settings Page Design and Creation
Max	- Login Page and logout button in settings page
”	- FitBit login pages and Android deep link
”	- FitBit data fetcher and parser
”	- Dashboard FitBit sections
”	- Internal Testing
”	- User personalisation from Google account
Michelle	- General UI design
”	- Reaction Time Test
”	- External Testing
Mika	- UI Design of the App
”	- User Survey
”	- Reaction Time Test
Tom	- Development Setup and GitHub Authority
”	- User Surveys Data Collection
”	- Notifications for Surveys, Daily Summaries and Emergencies
”	- Room Database
”	- Dark Mode

6.2.2 Report Authorship

Section	Subsection	Contributor	Word Count
Introduction	Problem Statement	Michelle	302
”	Solution	Michelle	136
”	Collaboration	Michelle	98
”	Integration with An Existing Prototype	Michelle	73
”	Purpose of New Prototype	Michelle	100
”	Product Description	Michelle	248
Design	Terminology (EMA)	Mika	243
”	Terminology (PVT)	Mika	194
”	Early-Stage Design Process	Mika	452
”	Prototyping on Figma	Mika	101
”	” (Homepage)	Mika	121
”	” (User Survey)	Mika	506
”	” (Reaction Time Test)	Mika	332
”	” (Data Dashboard)	Lewis	970
”	” (Settings Page)	Lewis	468
Development	Development Setup	Tom	602
”	User Surveys	Tom	569
”	Reaction Time Tests	Mika	952
”	Notifications	Tom	1250
”	Data Dashboard and Daily Summary	Lewis	948
”	FitBit Integration	Max	1614
”	Storing the Data	Tom	315
”	Login Page	Max	467
”	Settings Page	Lewis	581
Testing	Internal Testing	Max	1339
”	External Testing (Ethics)	Michelle	1620
”	” (Feedback)	Tom	613
Spec Outcomes	Technical Objectives	Lewis	616
”	Stretch Objectives	Max	958
”	Gantt Chart Comparison	Tom	393
Management	Project Planning	Tom	361
”	Division of Responsibilities	All	0
”	Risk Assessment Outcomes	Mika	810
”	GDPR and Ethics Compliance	Michelle	419
Conclusion	Conclusion of Project	Mika	299
”	Future Work	Max	626
Total			$\approx$ 19915

6.2.3 Poster and Presentation Contribution

Poster

Section	Contributor
Introduction	Michelle
User Surveys	Tom
Dashboard	Lewis
Reaction Time Tests	Mika
FitBit Integration	Max
Notifications	Tom
Settings	Lewis
Feedback and Future Ideas	Tom

Presentation Slides

Section	Contributor
Introduction	Michelle
User Surveys	Tom
Reaction Time Tests	Mika
Notifications	Tom
Dashboard	Lewis
Login Page	Max
Settings	Lewis
FitBit Integration	Max
User Feedback	Mika
Review of Project Objectives	Lewis
Gantt Chart	Tom
Conclusion	Michelle

6.3 Risk Assessment Outcomes

As the project is coming to end, we would like to dedicate this section of the report to reflect on the risk assessment plan that we had drafted at the beginning of the project. A screenshot of the risk assessment is shown below in Figure 6.1. At the start, we took a systemic what-if approach to identify any potential threats and risks that may negatively affect the outcome of our group design project. The first week of being assigned as group members also saw us having a sit-down together to discuss each other's strengths and weaknesses, as well as to identify opportunities and threats/risks that may arise from that. From there we were able to work out a few of the potential failure modes, and then proceed with evaluating them one by one. To do this, we used the severity, exposure, and probability (SEP) risk assessment method to evaluate and prioritise the risks we had identified. The three factors are briefly explained below:

1. **Severity:** This refers to the level of impact that risk would have on the project. It is typically calculated based on the size of the damage or loss that would result from the risk occurring.

2. **Probability:** This is the possibility that the risk will occur. It can be quantified by the likelihood of the risk occurring according to industry experts, or in terms of how frequently a particular risk had occurred in projects similar to ours.

3. **Exposure:** This describes the amount of risk to which the project is subject to. It is typically calculated based on the probability of the risk happening multiplied by the severity of the risk shall it occur.

*P = Probability (1-5); S = Severity (1-5); RE = Risk Exposure (Probability * Severity)*

Risk	P	S	RE	How to Mitigate?
Team member gets ill and cannot complete work according to schedule	2	4	8	Assess current workload situation, redistribute workload and maintain close communication with team member.
Covid spike causes loss of in person meeting and testing	2	4	8	Meetings can be conducted online and testing can be done on the team members or members of their households.
Work is lost due to computer failure	2	2	4	All the team's work will be centralised on online services such as GitHub and Google Drive so that any loss should be minor and localised to that one person.
Modifications are made to project requirements	3	4	12	Re-evaluate current plans and make changes to the Gantt Chart to account for the new project requirements.
Lack of knowledge and understanding of the frameworks required for mobile app development	4	3	12	All members should perform some reading of the relevant documentation to ensure only correct and latest technologies are used for developing the app.
Work runs over planned time	4	4	16	The team has clearly defined its core goals and the stretch objectives so if we are short on time the non-essential stretch objectives can be sacrificed.
Lack of engagement and communication with the supervisor and customer	2	4	8	The team should aim to meet the supervisor and customer at least bi-weekly to maintain synchronised about the project's progress. If engagement is poor then the team should contact the module lead to express any underlying concerns and be prepared to make modifications to the project specification.
Unable to fix certain bugs	3	4	12	Revert to a previous version using a version control tool such as Git and work from there.
Late approval of certain documents (ETHICS)	2	2	4	If ETHICS application is delayed/ unsuccessful then testing can be done on ourselves and other students and then we only need to abide by the universities health and safety regulations.

Figure 6.1: Our risk analysis table from the start of the project.

Evaluating each of the risks based on these factors resulted in the table shown above. Looking at the risk exposure (RE) scores, we deduce that we were most concerned about the project work running over the planned time as we had written in our Gantt Chart, with an RE score of 16. Besides that, there were three other risks we had identified that had an RE score of 12. These were the risks of being unable to fix some bugs in the code, modifications being made to the project requirements, and the lack of experience with the frameworks used to build mobile apps.

Now that the bulk of the project is completed, looking back at our risk assessment, it is rather comfortable knowing that we were able to avoid the top 4 risks with the highest RE throughout the timeline of this project, with the slight exception that we weren't fully able to adhere to the Gantt Chart schedule for some tasks. To point out some, things like drafting and creating prototypes of the UI did exceed the planned time for it. However, it did not affect our other processes all too much as we managed to quickly pull ourselves back on track immediately after it was done. We were still able to achieve our core objectives without feeling too pressured about running out of time before the deadline. We were also fortunate that no modifications had to be made to the project requirements, and that working with Android Studio didn't present us with problems that were exceedingly challenging to overcome. It was a gradual process for us to familiarise ourselves with Android Studio and some of the frameworks, but it surely did not turn out as bad as we imagined it to be in our risk assessment table.

We must shine a light on the risk of getting a late approval or the rejection of certain documents, however, as this was something that we all had overlooked when performing our risk analysis. As highlighted in Chapter 4.2.2, our team had struggled with getting our ETHICS application approved due to some slight confusion with the title of the project and the intent of the study. Looking back, a lot of time was spent revising the application form and resubmitting it again. This stripped away parts of the day that could be spent developing the app towards achieving our core goals as well as our stretch goals. It was only after three attempts we were finally able to get approval from the ETHICS board, but by that time it was too late to carry out the app tests on the OT students, preventing us from hitting one of our technical objectives. This was a big disappointment for us, especially now that we had reviewed our risk analysis from the start of the project – an RE score of 4 does not match and depict well the actual severity and exposure that it has caused us. The late approval of the ETHICS form was underestimated and overlooked, and this is something that we could all learn from when doing projects in the future.

6.4 GDPR and Ethics Compliance

The University of Southampton has strict policies, processes and guidance that must be adhered to for the University to be compliant when it comes to data protection. The Data Protection Act(DPA) is a law designed to protect personal data stored on computers or in an organised filing system. The Information Commissioner's Office(ICO), describes personal data as information about any particular living individual. This could include anyone, including a customer, client, employee, partner, member, supporter, business contact, public official or member of the public[29]. Personal data does not have to be 'private' information and if the information includes information that could make an individual identifiable, it still counts as personal data. This includes name, location details, email address, date of birth, online identifiers such as mobile device IDs or IP addresses and any other identifiers including physical, genetic, economic, mental or cultural. Special category personal data covers more sensitive information which includes: race or ethical origin, political origin, religious beliefs, Trade Union memberships, health data and sexual

orientation [31]. The ICO also states that the law applies to any 'processing of personal data'. Processing refers to anything to do with data including collecting, recording, storing, using, analysing, combining, disclosing, or deleting it.
It is University policy that ANY study involving human participants, questionnaires, lab studies, field observations and more, must be submitted for ethics review. As such in our project, we needed to collect feedback in the form of a survey once the app was complete to test its usability and functionality of the app. This process included the collection of some personal data such as their names and university email addresses in the consent form given to participants as participants had to be willing to partake in the testing of our app. Therefore, to be compliant with the University's Data Protection Policy, we submitted an ethics application to demonstrate how informed consent will be communicated, to identify potential risks and demonstrate how these risks will be addressed, and to establish how data would be gathered and protected. The university also suggests appropriately creating, storing, using, sharing, archiving and destroying personal data to help with compliance by using the following methods: using password-protected computers and documents, encrypting personal information, making backup copies and securing them securely, logging off or lock your computer, report suspicious cyber incidents and shred confidential waste. The process of submitting the ethics application is detailed in the external testing in Chapter 4 of this report.

Chapter 7

Conclusion

7.1 Conclusion of Project

In conclusion, this project has overall been a success in achieving its most of its objectives. Throughout the course of this project, the team was able to develop an app that would altogether allow a user to have a bird's eye view on the activities and circumstances that may contribute or correlate with their fatigue. All the processes – such as the user survey, reaction time test, integration with a FitBit, custom notification system, added user customisability, and dashboard – all come together towards achieving our main objective of helping users track their fatigue levels. The hard work and dedication of all team members has resulted in this positive outcome.

It should be noted that this project also served as a learning experience for all of us in the team. Along the way, we ran into a number of obstacles, but through our ability to work together and maintain in close communication at all times, we were able to overcome them. We have all grown in our understanding and experience of working in collaboration with an academic customer. In its entirety, it has been a valuable experience for all of us from the start, and we are proud of what we were able to achieve together. We would like to express our appreciation to our customer Dr Leisle, our supervisor Dr Mark Weal, and our second examiner Dr Mercedes Arguello Casteleiro, for working with us on this project.

While we are satisfied with the outcome of the project, we certainly recognise that there is room for improvement and areas where we could have done better. In the next and final section of this report, we have highlighted the areas where changes and further development can be made in the future. We will strive to implement them in our future endeavours.

7.2 Proposal of Future Work

There are a few areas within this app that could be expanded upon were we given more time or can be done by a future group working on this topic.

The most obvious area of extension would be to add support for more potential wearable devices to be added. This may be done by either adding integration for other wearable fitness devices besides FitBit, such as Garmin or an Apple watch or even designing a wearable from scratch that is designed specifically for the fatigue tracking app. Another option that sticks with the theme of getting more data into the app would be to try to integrate the app with other common apps used to track various fitness and activity metrics such as "My Fitness Pal" or "Strava" each of which have millions of users ¡add source¿ and would have the added benefit of keeping all the tracking within the phone removing the need for any wearable devices. This could make any research using the app more convenient and save money for the researchers as they don't have to make sure everyone taking part in The study has one of the wearables.

Another area of improvement that immediately stands out is to work with the customer to see what data they wish to have in the app from FitBit. This is because the data the app currently fetches from FitBit is just a couple of categories as a proof of concept as this was a stretch goal that was worked on towards the end of the Christmas break. Therefore, there wasn't an opportunity to have a consultation with the client about what information they wanted as unlike the survey and such for which we received very detailed instruction on exactly what survey they wanted in a high level of detail we had no instruction on the wearables side.

A topic that would also be good for extension would be to create some sort of server system that could store all of the survey result data that the researchers could access. This could make it a lot easier for them to gather data and allow them to monitor the participant's process while the trial is ongoing. This would also make life easier for the participants as once they got set up for the trial they might not ever have to meet the researchers again in person which lifts a lot of the geographic restraints normally associated with running the trials. If this was combined with the last potential piece of future work of integrating this app with other exercise tracking apps then this would lift both the equipment restriction and the location restriction which could make research using the app incredibly scale-able. The only thing that would limit study size at that point would be finding enough willing volunteers.

If there is future work done in this area one area that could also be expanded could be the survey format itself. At the moment it is the same survey every single time which may cause participants to get bored and may harm participation and engagement levels over the long term. There would be multiple ways to do this such as having a larger set of questions that are only asked a subset of each time. The app could also then be designed to ask the questions which get the most varied answers from the user. This is because these questions are likely to be the ones that both keep the user the most engaged and provide the most information for the researchers as answering the same question, in the same way, numerous times both disengages users and don't give the researchers much to work with.

Bibliography

[1] IEEE. "IEEE Standard Glossary of Software Engineering Terminology". In: (1990), pp. 7–74. DOI: https://doi.org/10.1109/IEEESTD.1990.101064.

[2] G. Kinsella et al. "Everyday memory following traumatic brain injury". In: (1996).

[3] Sylvia Loh et al. "The validity of psychomotor vigilance tasks of less than 10-minute duration". In: (2004).

[4] R.S. Boneva et al. "Higher heart rate and reduced heart rate variability persist during sleep in chronic fatigue syndrome: A population-based study". In: (2007).

[5] Julian Lim and David F. Dinges. "Sleep Deprivation and Vigilant Attention". In: *Annals of the New York Academy of Sciences* 1129 (May 2008), pp. 305–322. DOI: 10.1196/annals.1417.002. URL: https://nyaspubs.onlinelibrary.wiley.com/doi/full/10.1196/annals.1417.002.

[6] Saul Shiffman, Arthur A Stone, and Michael R Hufford. "Ecological momentary assessment". In: *Annual review of clinical psychology* 4 (2008), pp. 1–32. DOI: 10.1146/annurev.clinpsy.3.022806.091415. URL: https://www.ncbi.nlm.nih.gov/pubmed/18509902.

[7] J.L. Mathias et al. "Prospective and declarative memory problems following moderate and severe traumatic brain injury". In: (2009).

[8] W. Ament Castell et al. "Exercise and Fatigue". In: (2009).

[9] Bruce M. Hood. *The Science of Superstition : How the Developing Brain Creates Supernatural Beliefs*. 2010.

[10] In-Soo Lee et al. "Number of lapses during the psychomotor vigilance task as an objective measure of fatigue". In: *Journal of clinical sleep medicine : JCSM : official publication of the American Academy of Sleep Medicine* 6 (2010), pp. 163–8. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2854704/.

[11] B.D. Castell et al. "Cognitive behavioral therapy and graded exercise for chronic fatigue syndrome: A meta-analysis." In: (2011).

[12] Mathias Basner and David F. Dinges. "Maximizing Sensitivity of the Psychomotor Vigilance Test (PVT) to Sleep Loss". In: *Sleep* 34 (May 2011), pp. 581–591. DOI: 10.1093/sleep/34.5.581.

[13] Mathias Basner and Joshua Rubinstein. "Fitness for Duty: A 3-Minute Version of the Psychomotor Vigilance Test Predicts Fatigue-Related Declines in Luggage-Screening Performance". In: *Journal of Occupational and Environmental Medicine* 53 (2011), pp. 1146–1154. URL: https://www.jstor.org/stable/45009777.

[14] Michael Shermer. *The believing brain: from ghosts and gods to politics and conspiracies ; how we construct beliefs and reinforce them as truths.* 2011.

[15] Rolf Dobelli. *The Art of Thinking Clearly: Better Thinking, Better Decisions.* 2013.

[16] Centers for Disease Control and Prevention. "Traumatic brain injury surveillance report". In: (2019).

[17] Marie T Williams et al. "Compliance With Mobile Ecological Momentary Assessment of Self-Reported Health-Related Behaviors and Psychological Constructs in Adults: Systematic Review and Meta-analysis". In: *Journal of Medical Internet Research* 23 (Mar. 2021), e17023. DOI: 10.2196/17023.

[18] S. Dixon. "Main social networking site preference in the United Kingdom (UK) 2020, by age". In: (2022).

[19] Android. *Android Settings Design Guidelines.* URL: https://source.android.com/docs/core/settings/settings-guidelines.

[20] Apple. *Apple Watch SE Features.* URL: https://www.apple.com/uk/apple-watch-se/.

[21] Apple. *Shop Apple Watch.* URL: https://www.apple.com/uk/shop/buy-watch.

[22] FitBit. *Developer Intraday API guidance.* URL: https://dev.fitbit.com/build/reference/web-api/intraday/.

[23] FitBit. *FitBit Web API Explorer.* URL: https://dev.fitbit.com/build/reference/web-api/explore/.

[24] Fitbit. *Fitbit Store.* URL: https://www.fitbit.com/global/uk/products/trackers.

[25] Google. *MediaRecorder - Android Developers.* URL: https://developer.android.com/reference/android/media/MediaRecorder.

[26] Google. *Provide custom back navigation - Android Developers.* URL: https://developer.android.com/guide/navigation/navigation-custom-back#:~:text=All%5C%20Android%5C%20devices%5C%20provide%5C%20a,button%5C%20or%5C%20a%5C%20software%5C%20button.

[27] Google. *Save data in a local database using Room - Android Developers.* URL: https://developer.android.com/training/data-storage/room.

[28] Headway. *Acquired Brain Injury: The Numbers Behind the Disability.* URL: https://www.headway.org.uk/about-brain-injury/.

[29] licensed under the Open Government Licence v3.0. Information Commissioner's Office [Guide to Data Protection 14th October 2022]. *Some basic concepts*. URL: https://ico.org.uk/for-organisations/guide-to-data-protection/introduction-to-dpa-2018/some-basic-concepts/.

[30] Jovan. *Smartwatch Market Share: The World's Top Brands*. URL: https://kommandotech.com/guides/smartwatch-market-share/.

[31] University of Southampton. *Data Protection and GDPR*. URL: https://www.southampton.ac.uk/isolutions/staff/data-protection-and-gdpr.page.

[32] Julie Yuan. *authentication on Android embedded browsers*. URL: https://developers.facebook.com/blog/post/2021/06/28/deprecating-support-fb-login-authentication-android-embedded-browsers/.

Appendix A

Original Project Brief

Each year in the UK over 300,000 are admitted to hospital because of a brain injury. Around half will experience troublesome and persistent fatigue that prevents their resumption of work, social and leisure activities. Managing fatigue and the impact of fatigue on daily life is difficult, partly because fatigue fluctuates and can be difficult to predict, but also because people with brain injury may have cognitive challenges that affect their ability to keep track of and learn about their fatigue. To address this problem, we developed a working prototype app (android) that was tested by people with brain injury. The prototype included repeated surveys where people report their fatigue and activity at intervals across the day, sampled ambient noise and physical activity transitions, and delivered a short reaction time test (as an indicator of mental fatigue). A new prototype is needed to address design issues and further develop the app. These might include: to integrate data from external devices such as Fitbits, to provide daily summaries of the users survey responses, reaction time scores, and physical activity and sleep data from Fitbit, to allow some customisation by the user, to provide an enhanced short reaction time test to increase user engagement, track environmental information such as noise, possibly capture phone use data that might indicate fatigue – a slowing of response or increased errors on tasks.

Appendix B

Electronic Appendix Contents

B.1 Brain Fatigue App Code

Java files are contained in app/src/main/java/com/example/brainfatigueapp. Every .java file has an equivalent .xml file contained in app/src/main/res/layout/

- DashboardActivity.java
- DashboardFitBit.java
- DashboardLeftFrag.java
- DashboardPopupFrag.java
- DashboardRightFrag.java
- DaysActivity.java
- EmergencyService.java
- FatigueDatabase.java
- FitBitAPIHandler.java
- FitBitLoginActivity.java
- HomeActivity.java
- LoginPageActivity.java
- MainActivity.java
- NotificationReciever.java
- Reaction.java
- ReactionDao.java
- ReactionTimeTest1Activity.java
- ReactionTimeTest2Activity.java
- Setting.java
- SettingsActivity.java
- SettingsDao.java
- SummaryReciever.java
- SurveyEndActivity.java
- SurveyMiddle1Activity.java
- SurveyMiddle2Activity.java
- SurveyMiddle3Activity.java
- SurveyMiddle3aActivity.java
- SurveyMiddle3bActivity.java
- SurveyMiddle3cActivity.java
- SurveyMiddle3dActivity.java
- SurveyMiddle3eActivity.java
- SurveyMiddle3fActivity.java
- SurveyMiddle3gActivity.java
- SurveyMiddle4Activity.java
- SurveyMiddle4aActivity.java
- SurveyMiddle4bActivity.java
- SurveyMiddle4cActivity.java

- SurveyMiddle4dActivity.java
- SurveyMiddle4eActivity.java
- SurveyMiddle4fActivity.java
- SurveyMiddle4gActivity.java
- SurveyMiddle4hActivity.java
- SurveyMiddle5Activity.java
- SurveyMiddle6Activity.java
- SurveyMiddle7Activity.java
- SurveyMiddle8Activity.java
- SurveyMiddle9Activity.java
- SurveyResult.java
- SurveyResultDao.java
- SurveyStartActivity.java
- ViewStateAdapter.java

Contained in app/schemas/com.example.brainfatigueapp.FatigueDatabase/

- 6.json - Latest DB Schema

B.2 Customer Documents

- Dr Leisle's Thesis - Development of an Ecological Momentary Assessment of Fatigue after ABI using Sequential Exploratory Design
- Dr Leisle's Article - Experiences of Fatigue in Daily Life of People With Acquired Brain Injury: A Qualitative Study
- Brain Fatigue Survey Design
- Tracking Fatigue After Brain Injury Presentation
- Ecological Momentary Assessment - Shiffman

B.3 Ethics Documents

- Ethics Application Form
- Activity Questionnaire
- Consent Document
- Ergo Application (with revised versions)
- DFA Document
- Risk Assessment Document
- Participant Information Sheet

B.4 Project Management Documents

- Gantt Charts

B.5 Survey Documents

- Questionnaire PDF
- Questionnaire Results

B.6 System Design Documents

- Version 1 UI Design
- Version 2 UI Design

B.7 Additional Appendices

- App Screenshots
- Report Documents