Anxiety Manager (incorporating Wearable Technology) Final Year Project Report

DT228
BSc in Computer Science

Katie Fitzgerald
Damian Bourke

School of Computing
Dublin Institute of Technology

13th April 2018

Abstract

This project aims to design and develop a mobile solution to give individuals with anxiety an innovative way to track their anxiety attacks and learn about their own anxiety. Currently, mental health tracking solutions are unable to compete with technology in people's daily lives. This project embraces that technology to replace the outdated pen and paper tracking methods, with questionnaires contained in an Android app. Along with automatic tracking of anxiety attacks, through sensors monitoring changes in heart rate and sweat levels (common anxiety symptoms). These sensors reduce the pressure on users to document their attacks as soon as they happen, rather allowing them to fill out questionnaires in their own time.

From this tracking, users can obtain visual insights into their own anxiety attacks, which traditional methods lack. Insights include most common location, the correlation between mood and reaction and the frequent triggers of anxiety attacks. Anxiety Manager also allows healthcare professionals to receive these insights about their patients.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

Katie Fitzgerald

Katie Fitzgerald

13th April 2018

Acknowledgements

I would like to express my deepest appreciation to all those who provided me with the possibility to complete this project. A special gratitude I give to my mentor Damian Bourke whose valuable advice and encouragement, helped me to coordinate my project.

Furthermore, I would also like to acknowledge with much appreciation my family and Andrew for the constant support during the preparation of my project. You kept me motivated and calm when the stress got the better of me, which I am genuinely grateful for.

Finally, I want to extend my gratitude to the lecturers of DIT School of Computing who were my biggest inspiration and empowered me endlessly during my degree.

Table of Contents

Table of Figures

Chapter 1 Introduction

1.1 Project Overview

The purpose of this project is to develop a system, for users of different IT skill levels to track the physical and non-physical characteristics associated with anxiety attacks. The project challenges the outdated current process of tracking anxiety attacks. This includes physical symptoms, moods, thoughts and reactions to worrying situations often tracked in a diary or anxiety workbook. This method can cause even more anxiety for people as writing this information down after an anxiety attack (which often occurs in public) is quite invasive.

The developed system incorporates a wearable sensor and a mobile application. The sensor monitors changes in the user's physical symptoms, such as heart rate or sweating. The sensor allows autonomous tracking of anxious situations. This means users can track the anxiety attack later in their own time. Along with sensors, the developed system includes a mobile application, with questionnaires to log anxiety attacks and insights into trends associated with the user's anxiety. The mobile application will also have a profile for healthcare professionals with access to their patient's trends.

This project minimises the added stress that comes with tracking anxiety. It uses minimal inputs, emoticons and numbers to portray the information needed to successfully track anxiety attacks. It also gives the user a visual representation of their anxiety. When a person learns about their own anxiety, it gives them the knowledge to break their anxiety cycle. Giving healthcare professionals access to user's anxiety trends will enable them to provide the right tools to their patients to deal with their anxiety.

"Around 1 in 6 people in Ireland will experience a mental health problem like anxiety each year" [1]. Anxiety can be debilitating. Mental health issues continually impact society and the way people live. By taking advantage of technology already present in people's live anxiety can be tackled and mental health issues can be given a better understanding.

1.2 Project Objectives

The objective of this project is to develop a non-invasive, easy to use, mobile system that provides understanding to the user about their anxiety and, will overcome the issues associated with current tracking approaches. To accomplish this, a system must be developed that users of all varying IT skills find easy to use. The system must produce autonomous ways of sensing anxiety attacks. The system must provide effective accessible ways to logging information about anxiety attacks. The system must take that information and convey it back to users in a way that is both comprehensive and explanatory.

To attain this objective, the following goals must be met:

- Acquire a profound understanding of anxiety and what information is vital to record when tracking anxiety.
- Research appropriate mobile systems, while keeping in mind the demographics of the system.
- Research wearable sensors that can monitor physical symptoms associated with anxiety.
- Research existing solutions available that help users track anxiety.
- Based on research, design and propose a system that is easy to use and minimal.
- Produce prototypes from the design phase and receive feedback from users on the designs.
- Develop the system.
- Test the system.

1.3 Project Challenges

1.3.1 Introduction

As with any large-scale project, there are associated risks and issues. Usually they are related to lack of resources such as time or technologies. This section will identify the risks of this project. They include issues with time, sensors, technologies, and privacy concerns.

1.3.2 Time sensitivity

Firstly, one risk associated with this project is time running out. This project must be researched, designed, developed and tested over an academic year. It is possible that not all planned development and testing will be carried out. To overcome this challenge, priority must be given to user requirements that achieve the objective quickest.

1.3.3 Sensor connectivity

Another risk associated with this project is that the chosen sensors will not work as predicted. The intended use of the sensors is to monitor user's physical symptoms of anxiety such as sweat or heart rate. They will detect changes from the "normal" sweat amounts or heart rate for the given user. Possible problems will include:

- Sensors could be too sensitive to accurately measure changes
- The generally perceived "normal" rate of physical symptoms, might not fit all users.
- Sensors fail to receive any change in users.

Due to the time restriction included in this project, data from sensors could have to be simulated to ensure the proposed idea is validated.

1.3.4 New technologies

Learning new technologies can be time-consuming and challenging. It certainly comes with risks. New technologies may change the intended plan for the project or cause huge delays. As technologies are researched, these risks should be accounted for.

1.3.5 Privacy

The sensitive nature of the project subject poses an enormous risk to the success of this system. Users may not be willing to share sensitive information about their mental health. They may feel overwhelmed by wearing visible sensors that will be monitoring them. In turn, this feeling may cause inaccurate readings from sensors. As well as sensors receiving sensitive data, the questionnaire within the application also requires users to share personal information. The questionnaire asks them to share their thoughts and their moods. It is essential that users are aware that their information will only be available to their healthcare professional with their permission. If users do not wish to share their data, the author will track their own anxiety for testing purposes.

1.3.6 Conclusion

It can be said that there are few identifiable risks relating to this project. The main risk is due to the technology not working as intended, but consideration will be made for this. As mentioned, because of the time restriction of this project, there will not be time to iterate on the decision made about chosen sensors. Alternatively, data will be simulated to show that the application could track a user's anxiety given the correct conditions. As well as that, developing a coherent plan will help eliminate any issues relating to time running out and executing features based on the priority will reduce risks. Adhering to privacy standards will be vital for this project and ensuring clarity is given about what the data is being used for will eliminate any uncertainty for the user.

1.4 Report Structure

Research

This chapter explores the background research about anxiety and tracking, as well as a look at existing technologies similar to the project. Following that, the research undertaken to decide the most suitable technologies for the project is discussed. That research is later used to curate the user requirements for the system.

Design

This chapter explores the chosen methodology for the project and how that decision came about. Observing that, this chapter reviews the detailed design of the system and the iterations taken to come to the final design.

Architecture and Development

This chapter breakdown the development of the system following the technical architecture design, front-end, back-end and middle layer. It also considers the challenges and workarounds experienced throughout the various levels.

System Validation

This chapter examines the usability, requirement and sensor tests that were used to validate the system and an acute evaluation following Nielsen's Heuristics and a general analysis.

Project Plan

This chapter discusses the project plan and how it changed over the course of the project. It also explains the future for Anxiety Manager.

Conclusion

This chapter reflects on the project whole process, the learning experience during the course of the project and what would be changed if the project was undertaken again.

Chapter 2 Research

2.1 Introduction

As mentioned, the research portion of this project is critical to accomplishing the objective. A broad understanding of anxiety needs to be expressed before decisions are made regarding the system. Research needs to be gathered on symptoms associated with anxiety (to prompt research about sensors). As well as, research on tracking anxiety attacks with technology and how to present the data visually and effectively to individuals.

For the system to be successful, how users interact with it is important. The objective of the project is an easy to use application for a variety of users - IT and non-IT savvy. Appropriate inputs will need to be researched based on this objective. This can be achieved by examining alternative solutions already in place for users with anxiety.

The system will involve an extensive amount of research around non-invasive sensors for those with anxiety. Similarly, the technology that allows the sensors to interact with the mobile application will need to be examined. The mobile application technology will need to reflect the systems demographic.

This chapter will explore all the groundwork concerning anxiety, tracking anxiety, existing solutions to anxiety tracking, and appropriate technologies for the system. It will also explore research concerning user interface elements that will be effective in the application.

2.2 Background Research

2.2.1 Introduction

For the developed system to successfully assist users to manage their anxiety, an extensive amount of research needs to be gained. A comprehension of the complete spectrum of anxiety, who experiences it and how it manifests must be found. As well of that, how anxiety should be tracked to achieve results, that allow individuals to become aware of their habits needs to be learnt. This section will explore the research undertaken to gain that comprehension.

2.2.2 What is anxiety?

Firstly, the question "what is anxiety?" needs to be answered to understand the need for this system. Anxiety can be described as the product of apprehensive behaviours. When humans experience emotions, such as fear, concern, or worry, they “behave in an apprehensive manner” [2]. The Merriam Webster dictionary defines anxiety as “an abnormal and overwhelming sense of apprehension and fear often marked by physical signs (like tension, sweating, and increased pulse rate), by doubt concerning the reality and nature of the threat, and by self-doubt about one's capacity to cope with it" [3]. Anxiety is a rational response to stressful, dangerous or new situations. For example, three out of four individuals experience some level of anxiety before public speaking [4]. When anxious feelings are present, a 'flight or fight' response is triggered and our bodies prepare to react. Hearts begin pumping blood to muscles quicker, to run away from danger [5]. Anxiety becomes a problem when there is no reason or context for feeling anxiety. Catherine Bolger of DIT counselling services described normalised anxiety during an interview in October 2017, as "having a context to place a worry or anxiety in, means it is looked at in a much more normalised fashion" [6]. Undergoing the same feeling as before public speaking, while going about a normal daily routine is not a habitual way to experience anxiety. Having our bodies respond to hypothetical fear or danger over an extended period is not safe. This establishes the need for some way of stopping this type of irrational fear. When this kind of anxiety without a valid reason happens multiples times, it can be described as a disorder. Extreme anxiety can be debilitating and cause people who experience it to be apprehensive about daily tasks. [7] Anxiety causes sufferers to always expect failure, danger or even disaster to the point of it inferring with daily life. Mental health issues affect millions of people in Ireland. With the rise of busy schedules and an increase in demands from people in personal and professional life, current solutions to overcoming anxiety are no longer a "one size fits all." Straightforward and diverse mobile solutions are needed more than ever to overcome mental health issues such as anxiety. This leads to the question, who is anxiety affecting? The answer should lead to the user base of this system.

2.2.3 Who is anxiety affecting?

Following the examination into what anxiety is, who anxiety is affecting needs to be known. As mentioned, anxiety does affect everyone in their life as it is an instinctive response to unfamiliar experiences. Anxiety disorders impact people in ways normalised anxiety doesn't. People with anxiety trait personalities are likely to experience anxiety. It is more likely that individuals with a family history of mental illness will be susceptible to anxiety, due to a chemical imbalance in their body [8]. Individuals who are being exposed to prolonged stressful or fearful states (such as relationship problems, financial stress, or work stress) will most likely develop anxiety disorders. Anxiety doesn't distinguish between age, background or social status due to the wide range of states that cause it. Because of the commonality and the generality of anxiety, the developed system must be geared towards every type of user. Meaning the application developed must aim to be accessible to users of every age, background, social status and IT skill level. Now by differentiating what type of people is being affected by anxiety, how anxiety is manifesting for these individuals needs to be learnt.

2.2.4 How does anxiety manifest?

It is now known anxiety can affect any member of a population. How it manifests is on an individual basis needs to be explored. Anxiety can manifest through 'anxious episodes' more often called anxiety attacks or panic attacks, or through mood swings and unexpected behaviour. An anxiety attack begins when an individual notes some unusual feeling in their body. This can range from a physical symptom or to racing thoughts. From there, they become hypersensitive to the situation they are in. This is when the mind unconsciously monitors the body and notices any sort of change [9]. When a change happens, either physically or emotionally, the situation provoking it is amplified and seems worse than it really is, creating a cycle of bad feelings.

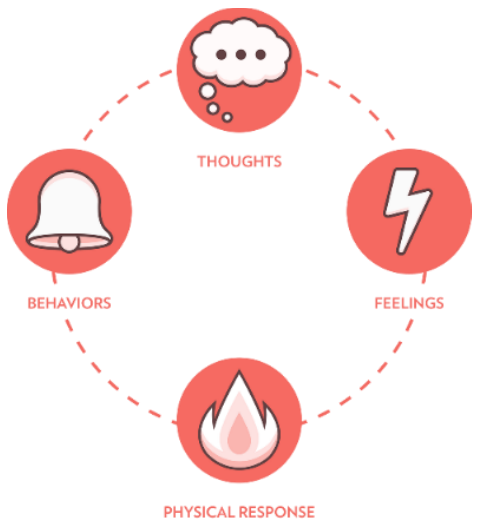


Figure 1- Anxiety Response Cycle [9]

There is a general method used by counsellors and healthcare professionals to describe the cycle of anxiety episodes, shown in Figure 1. The cycle usually begins with an unsolicited unprovoked physical response to a situation. These responses can include (but is unlimited to) shortness of breath, feelings of warmth, shaking, or pains in stomach. These physical feelings begin to alarm individuals and they assume something is wrong. Subsequently, negative thoughts begin and are usually related to a lack of capacity to deal with the situation. Sequentially, negative thoughts can cause the physical symptoms to become more pronounced manifesting the attack further. Finally, the individual is left with a decision to stay and fight against the negative thoughts, or leave the situation and get instant relief of symptoms. Leaving the situation is predominantly not an appropriate behaviour, as it does not train the individual to cope with their anxiety. It is necessary the developed system communicates this knowledge to users.

Anxiety also manifests itself in mood and behaviour. Individuals with anxiety become very irritable over time due to their body and mind being on constant alert for fear or danger. Some frequent behaviours demonstrated in people with anxiety include compulsion, nervous tics or sleep disturbance. As mentioned, avoidance behaviour is the most common response to anxiety attacks. This behaviour involves individuals going out of their way to avoid situations that maybe have pronounced anxious feelings before. Because of this, individuals become very isolated and their moods and behaviours suffer as a result. Catherine Bolger believes that anxiety can compel individuals to become very frustrated and exasperated, as it's not an understood disorder. Individuals with anxiety are seen to be overreacting about simple day-to-day situations [6], but realistically they tend to not be knowledgeable about what is causing these irrational feelings. It could be assumed people are unaware of the causes of their own anxiety due to a lack of tracking and monitoring solutions. Any system developed to tackle this issue needs to convey a personal awareness of anxiety to the user.

It is apparent from research that how anxiety presents itself in individuals varies but a lack of awareness of personal anxiety triggers is common. The type of tracking required to give this awareness to users through the system needs to be explored.

2.2.5 How do you track anxiety?

As discussed, it is very hard for an individual to see their anxiety trends by themselves. Tracking how every anxiety attack manifests gives way to the overall triggers of an individual's anxiety. Correct tracking of anxiety is the only way for a person to get a deep understanding of their own disorder due to its subjective nature. For tracking to be successful, "what's happening, the physical symptoms being felt and the associated thoughts" [6] must be taken note of consistently. Healthcare professionals or counsellors ask their patients to log anxiety attacks so they can get to the root of the underlying issue. An individual should set aside a dedicated time daily, to log mood and thoughts or track anxious episodes as they occur [10].

When designing a system for tracking anxiety, it must incorporate each element of anxiety in some way. The items that need to be noted when tracking an anxiety attack [5] [10]:

1. Location - where was the person?
2. Subject - what was the event?
3. Thoughts - what were the relating thoughts?
4. Physical symptoms - what were the physical feelings?
5. Behaviours - what were the actions of the individual?
6. Mood - how was the person feeling emotionally?
7. Reaction - what was the reaction to the anxiety attack?

An anxiety attack location can offer comprehension. The environment of an attack may be an unknown trigger to the individual. "Through regularly tracking panic attacks, you may notice that they often occur when you are in specific situations or events" [11]. Along with the subject or event, patterns may arise in attack occurrences. Anxiety can be a learned behaviour, therefore being exposed to locations or events that are causing anxiety may be the root of the problem. This verifies the need for location and subject tracking within the developed system.

What a person is thinking prior to or during an anxiety attack allows healthcare professional to get a perspective on the situation. An attack usually accompanies racing thoughts. Racing thoughts are fast, repetitive thought patterns about a subject or situation [12]. "My chest is beating rapidly, am I dying?", "I am sweating a lot, can everyone see I am sweating? Are they going to laugh at me?" or "I can't breathe properly, am I going to faint?". Sometimes anxious thoughts can be triggered by past situations that generated anxiety, or the current physical symptoms [6]. The thoughts associated with anxiety are hard to change, and according to Catherine Bolger "learning about patterns cuts the cycle of anxiety and decreases the worrying thoughts" [6]. This means the system needs to have a way of recording thoughts.

As mentioned, anxiety attacks can begin with physical symptoms. Physical symptoms of anxiety include:

- Unusual heartbeat (fast, pounding)
- Unusual breathing (fast, shallow)
- Feelings of warmth or hotness
- Excessive sweating
- Nervous stomach or nausea
- Headaches

Physical symptoms tend to lead to individuals fearing the outcome of these symptoms. For example, when people with anxiety feel an increased heart rate, they assume the worst is going to follow. Individuals can experience symptoms relating to anxiety without considering it to be a connection [6]. Physical symptoms can vary from person to person, but it is important individuals gather an awareness of how their body is responding to apprehensive or fearful situations [6]. From time to time, physical symptoms may not be present in a person with anxiety. Instead, as previously described, their behaviour is a product of their anxiety. Some very common behaviours that can indicate anxiety include:

- Loss of appetite
- Increased appetite
- Shaking (nervous tics)
- Frequent toilet trips

As physical symptoms and behaviours are the leading indicators of an anxious situation, it is necessary for them to be noted in the developed system. This will train users to pinpoint exactly what physical symptoms or behaviours are common for them and learn to recognise they are okay when they feel them.

By taking note of mood, a tremendous insight is given into a person's anxiety. Mood can communicate a general idea of how anxiety is affecting a mind. Mood and emotion play a role in issues of survival and involve cognition and behaviour, which is why anxiety heavily affects mood [13]. Mood will need to be addressed within the developed app as it is so substantially linked to anxiety. Anxiety alters the way the brain receives and sends neurotransmissions and the levels of hormones in a person's system. As a result, cause severe mood swings [14].

Moods are generally perceived to be states of feelings or emotion. Psychologist Robert Plutchik's Theory of Emotion states that there are eight basic emotions: joy, trust, fear, surprise, sadness, anticipation, anger and disgust [15]. From these eight basic emotions, he believes there is a spectrum of other emotions. In Figure 2, there is 'Plutchik's Wheel of Emotion' that helps visualise the spectrum of emotions and how they relate to one another. Emotions intensify from the outside to the centre of the wheel in Figure 2.

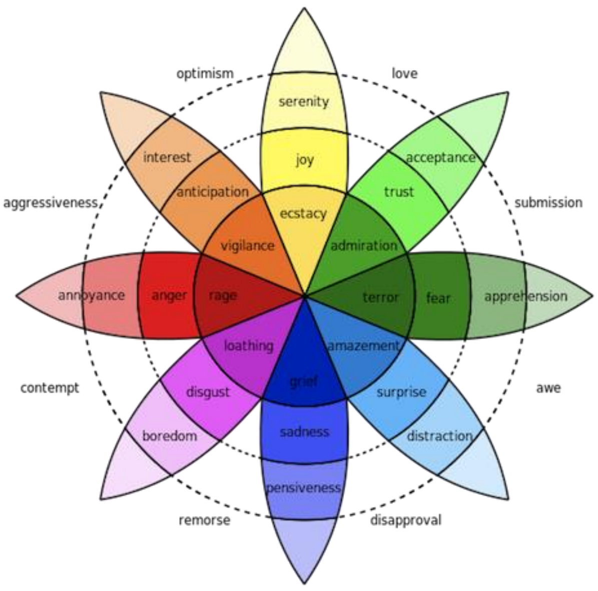


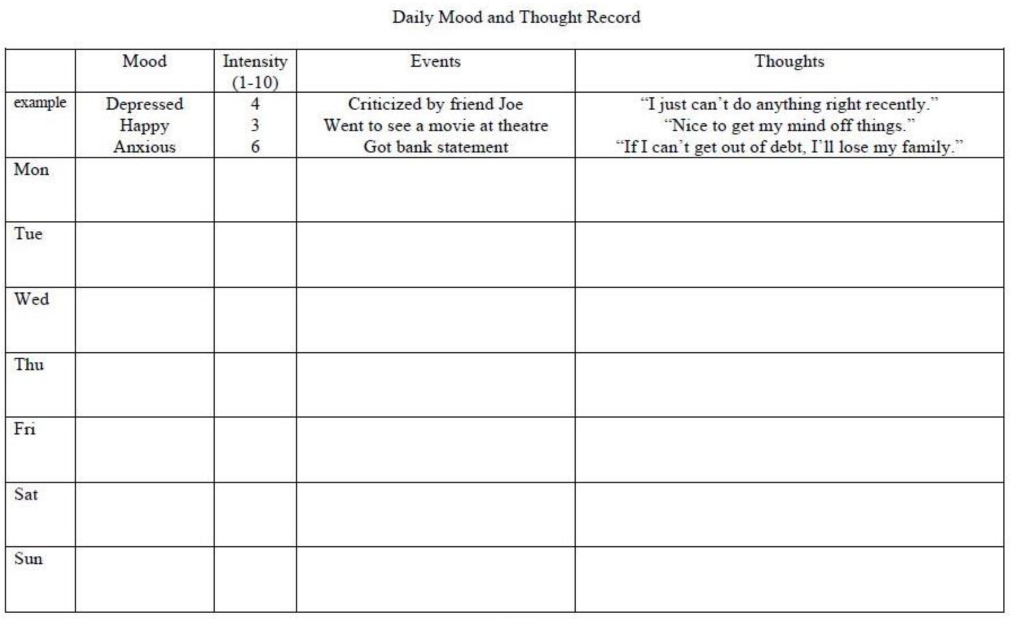
Figure 2 - Robert Plutchik's Wheel of Emotion [15]

This indicates a need for moods to be tracked not as linear feelings but, on a scale of how intensely they were felt during an attack. As well as that, there is an essential rule that accompanies tracking emotion and mood: "If left unchecked, emotions can intensify" [16]. This means that over time if moods that are being directly caused by anxiety are unobserved they will get worse and become 'normalised' for an individual. This reiterates the requirement to note mood when tracking anxiety attacks within a digital system.

The reaction to anxious situations is another crucial factor in highlighting an individual's anxiety to themselves. As previously mentioned, avoidance behaviour is an intuitive response to anxiety attacks. Records of this constant fleeing can give a way of demonstrating to an individual how they aren't coping with their anxiety. If an individual is constantly fleeing from anxious situations, they will likely never see any change in their mood or behaviour. "Negative reinforcement in the case of anxiety can be thought as avoidance." [17] This means every time an individual lets their anxiety take control of situations and they leave the situation, they are avoiding facing up to their fears and apprehension. For an individual to overcome their anxiety, they need to see the repercussions of their avoidance. The developed system must indicate to the individual that when they leave their symptoms are not changing and that there is no long-term reward for leaving, only a fleeting moment of relief [6].

2.2.6 Current Anxiety Tracking Solutions

The most common solution to tracking anxiety is 'pen and paper'. This solution generally includes an individual taking time out of their day to focus on written inputs. The individual tracks their physical symptoms, behaviours, thoughts and moods during their anxious attacks by writing them down in a diary or a workbook (usually given to them by healthcare professionals).



Daily Mood and Thought Record

	Mood	Intensity (1-10)	Events	Thoughts
example	Depressed Happy Anxious	4 3 6	Criticized by friend Joe Went to see a movie at theatre Got bank statement	"I just can't do anything right recently." "Nice to get my mind off things." "If I can't get out of debt, I'll lose my family."
Mon				
Tue				
Wed				
Thu				
Fri				
Sat				
Sun				

Figure 3 - Example of anxiety episode workbook [18]

Figure 3, shows the types of topics required of individuals to answer when manually tracking their anxiety using a workbook or diary. This allows a visual way for a person to cognitively become aware of their anxiety. This is an advantage of the pen and paper solution to tracking anxiety. Unfortunately, there a few disadvantages associated with this solution. One disadvantage is an individual must remember to keep the diary or workbook with them in case of an anxiety attack. Similarly, when attacks occur they can be overwhelming themselves. Having to remember to track episodes as they happen, by writing them down, can be difficult when anxiety consumes thoughts. A disadvantage of having to write anxiety down in a workbook or diary can be that it can trigger anxious feelings itself. If an individual needs to log an attack in a public forum, subsequently writing a very personal problem in that public forum can cause more anxiety [6]. Overall, it is most beneficial for people with anxiety who wish to track anxiety with pen and paper to track their symptoms in a private place. This avoids creating more issues from the tracking or running the risk of forgetting to track.

In recent years there have been more innovative answers to tracking general health and mental health issues, to avoid the stigma of tracking symptoms publicly. These solutions are mobile applications. Using mobile applications to track mental health issues, such as anxiety, avoids the disadvantages of pen and paper solutions. Mobile systems are adaptable to modern schedules and according to TIME Mobility poll, "84% of people couldn't go a single day without their mobile device in hand" [19]. Logging mental health matters through mobile devices grant users the freedom of when and where they can track. Personalised notifications and reminders on mobile devices can also tackle the problem of remembering to log issues. This restates the need for the developed system to be a mobile non-invasive private system.

2.2.7 Purpose of tracking anxiety

Having tracked anxiety, it is important to have a professional to help with a successful treatment of anxiety [51]. Tracking alone will not cure anxiety, but can only help identify triggers. The developed system only focuses on tracking of anxiety and finding patterns in attacks. The system should be used by anxiety users in conjunction with their healthcare practitioner or counsellor.

Counsellors should only have access to the following data about their users:

1. Sensed anxiety: when and where anxiety is felt and what way did they patient feel and react?
2. Anxiety cycle: what are the trends of the user's anxiety episode? Where are they happening mostly? How is their reaction impacting their mood? And what is provoking the most anxiety?

The counsellor will be able to access this information from their own mobile application. Users must consent to their counsellor view their data by selecting them through the system. From the data accessed, counsellors can provide suitable treatment based on trends.

2.2.8 Conclusion

Following the acute look at anxiety (including the symptoms, behaviours and moods associated) and the type of people who experience anxiety, a user persona can be established. As anxiety does not distinguish between age, this proposed system must be easy to use for all types of users who experience any form of anxiety. As well as knowing how anxiety manifests, permits the developed system to automate the tracking of anxiety. Tracking must be done consistently for anxiety trends to be examined, creating a need for daily reminders. For this project to uncover the root of individuals anxiety with tracking, the developed system must adhere to these tracking practices of physical symptoms, moods and behaviours. Now that an understanding of current tracking solutions has been established, those methods can be adopted by the developed system. To gain a view of how technologies are currently trying to tackle the problems with anxiety tracking, the next step is to examine some existing solutions.

2.3 Existing Solutions

2.3.1 Introduction

This project challenges the automating of tracking of an individual's anxiety to ensure the accurate anxiety management practices can be provided by healthcare professionals. Currently, there are few solutions to this problem. Some solutions directly related to this project, being mobile applications. As previously mentioned, traditional solutions for tracking come with obstacles. The following mobile applications enable users to monitor anxiety attacks without creating any other issues:

The Worry Box - a cognitive-behavioural therapy application for people who experience anxiety and worry, that allows users to learn how to recognise the importance of their anxiety and worry. It teaches users to "identify irrational thoughts" during anxious and worrying periods [20].

Self Help for Anxiety Management - a system that helps users understand and manage anxiety. It monitors what causes individual's anxiety and the associated thoughts and behaviours [21].

What's Up - an app that uses "some of the best Cognitive Behavioural Therapy and Acceptance Commitment Therapy methods to help you cope with Depression, Anxiety, Anger, Stress". [22] This application helps users become aware of negative thinking patterns and gives users methods to overcome them.

Each of these apps directly relates to overcoming the challenges of traditional tracking methods and will be evaluated based on Nielsen's Heuristics and their features.

2.3.2 Evaluation of Existing Technologies

The mobile applications will be evaluated based on these criteria:

1. Features
    a. What features are available?
    b. Do the features make users aware of their own anxiety trends?

2. Usability and Design - Nielsen Heuristics (NH) [23]
    a. Does the application keep users informed?
    b. Does the application observe real-world conventions?
    c. Do users have the freedom to undo actions and rectify mistakes?
    d. Is the vocabulary of the application consistent?
    e. Does the application require users to recall information to complete tasks?
    f. Does interaction speed increase for experienced users?
    g. Is the design minimal?
    h. Are error messages expressed in plain text that is easy to understand?
    i. Are help and documentation provided?

The Worry Box

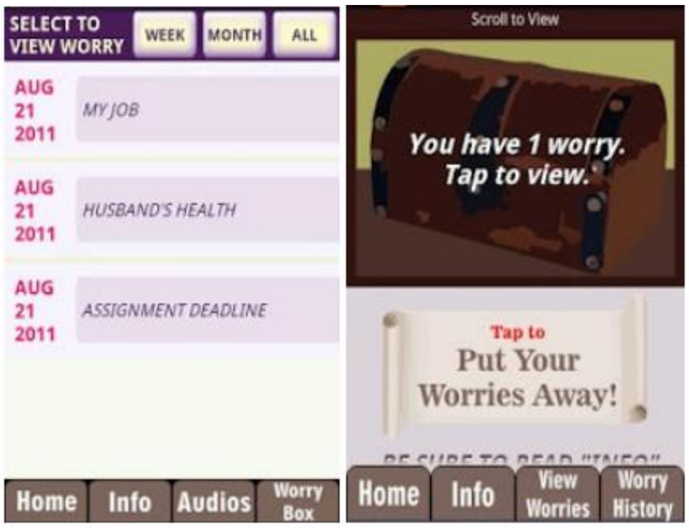


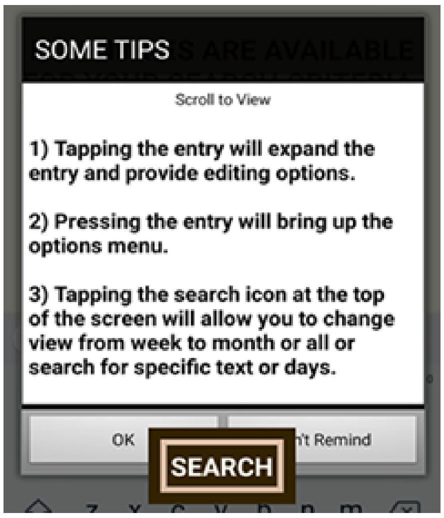
Figure 4 - The Worry Box application

The features of this application are:

- Worry questionnaire
- List of logged worries (Figure 4)
- Audio recordings of mindfulness
- Articles about cognitive behaviour therapy and managing stress

The worry questionnaire is relevant to the developed system from this project. The items logged include a title, description, importance of the worry rating and steps to cope with it. This indicates that a questionnaire may be the easiest way to retrieve information from users about their anxiety attack.

The Worry Box has good usability, but it can be frustrating to use at times. There are accidental exits and numerous pop-ups one after the other. It can be imagined that during periods of high anxiety utilising an application that causes frustration is not ideal. The Worry Box succeeds in giving constant feedback and freedom to its users during their time using the application. It informs them which part of the application they have navigated to or what is required of them to do in certain activities. There is always a way for the user to undo their actions and return to previous activities. A good example of useful visibility of the system is pop-ups. In this application, they are used to give users instructions on how to add worries to the app. The application also accomplishes good usability standards by speaking the user's language by applying real-world phrases to text. For example, the application defines 'What is Worry?' employing very common terms such as "anxious or upset" and "you can learn to use relaxation methods to help manage to worry". These are conventional and general terms known by wide demographics of users.


SOME TIPS

Scroll to View

1) Tapping the entry will expand the entry and provide editing options.

2) Pressing the entry will bring up the options menu.

3) Tapping the search icon at the top of the screen will allow you to change view from week to month or all or search for specific text or days.

Figure 5- Hidden pop-ups

The good usability of the application is shadowed by poor design and extensive bodies of text. In Figure 5, it is seen that the buttons to dismiss the pop-up are hidden behind a toast relating to a different action. This is something that needs to be avoided in the developed system. Despite the proper user control and freedom conventions, error prevention is not at the core of this application. User actions are not consistent across the application. For example, it is very easy to exit the application accidentally. The back button on the home activity closes the app and is situated where the menu icon generally is in mobile applications. While in other parts of the app, the back button returns to the immediate previous page. It appears that previous activities are not destroyed when users have fulfilled their tasks. To add confusion, error messages or instructions require a lot of reading or do not relate to the problem at hand. For example, when you open the worry log without having a previously logged a questionnaire, the page displays an error message stating: "NO ENTRIES ARE AVAILABLE FOR YOUR SEARCH CRITERIA. TRY DIFFERENT SEARCH CRITERIA (TOP OF SCREEN)" although a user has never carried out a search.

Overall, The Worry Box demonstrates a satisfying standard of NH. From The Worry Box, it can be learnt the developed system must give constant feedback to users. As well as that, grant users the freedom to undo their actions. The developed system must avoid massive amounts of text and poor error prevention. Error messages must be clear and concise. The features of The Worry Box that makes users aware of their anxiety are the worry questionnaire and the list of logged worries. These features should be adopted by the developed system.

Self Help Anxiety Management (SAM)

The features of the SAM application are:

- Anxiety questionnaire
- "Things that make me anxious" list
- Anxiety toolkit

- Anxiety tracker
- Social cloud
- Self-help exercises
- Anxiety help for right now
- Documentation
- Anxiety links

There is a vast range of features in this application. The anxiety questionnaire and tracker are great features to support users to become aware of their anxiety. The anxiety tracker indicates the trend in the users tracking with a line chart. This is a very effective way to highlight trends to the individual by not using text and something that could be incorporated into the developed system.

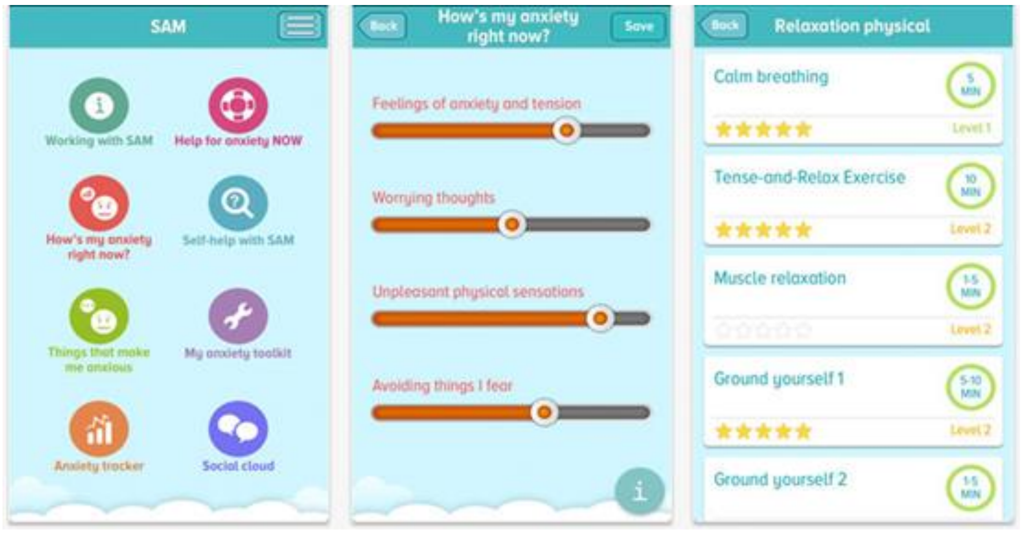


Figure 6 - Self Help for Anxiety Management application

Overall, SAM has great usability standards. It has a simple design that makes the application easy to navigate. As well as that there is no confusion for users. SAM achieves this by directly telling users where they are throughout the application with page titles and buttons with appropriate labels, Figure 6. Using page titles and, providing icon and button labels through the proposed application of this project will be necessary. Similarly, the system matches the real-world. It, much like Worry Box, doesn't use system specific terms. The target group of the application is young adults aged 15-25. It employs real-world terms such as "panicky", "stressed out" and "patterns in your anxiety" to explain the monitoring of anxiety in the application. That helps novice and expert users relate to the workings of the system. This is something that should be integrated into this project.

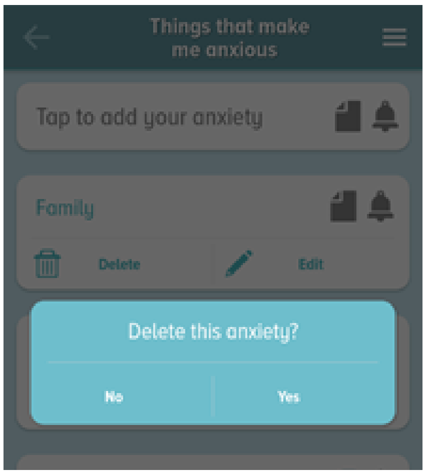


Figure 7- SAM confirm action screen

As well as having a good match between the system and the real world, there is an extensive amount of user control and freedom within the application. When a user navigates to most parts of the application, there is a back button to return them to the homepage or previous page. Likewise, there is freedom to delete and edit inputted data within the application. A user can edit and delete their added anxieties. Error prevention could be considered at the forefront of this application. There are restrictions on user inputs which avoids human error. As stated, users can remove any unwanted data they previously added but only after confirming the operation will the action be committed. I.e. When a user wishes to delete an entry of 'Things that make you anxious', the user is met with a pop up that says, 'Delete this anxiety?' with two buttons for 'No' and 'Yes', seen in Figure 7.

The only downfall of SAM is that users are required to recall a lot of information across the application. When a user adds an entry to their anxiety toolkit, they are given the instructions on how to remove the entry. This isn't exactly appropriate because removing entry means navigating to an entirely different part of the application. This increases the user's memory load. It is required of them to retain information for one part of the application from another part. As mentioned, it is crucial for mental health applications to not frustrate or overwhelm users, meaning this project must look at creating simplistic actions for users in the developed system.

The features that help users realise their anxiety trends in SAM are the anxiety questionnaire, "things that make me anxious" list and the anxiety tracker. These are key features needed in the developed system to fulfil the objective.

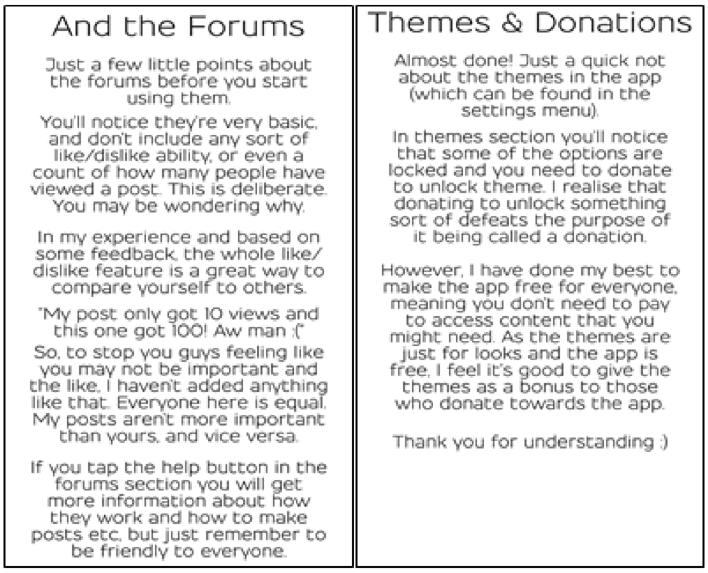
What's Up?

The features of this application are:

- Personal diary

- Habit Tracker
- Grounding games
- Coping Strategies
- Information about mental health
- Uplifting quote
- Catastrophe scale

What's Up is an application tailored for users with depression, anxiety, anger or stress. This system helps users become aware of their negative thinking patterns and implements methods to overcome them. The application also has a diary like SAM and The Worry Box for users to log thoughts and feelings together. What's Up demonstrates a high standard of usability. Although the design of the application is not minimal and there are excessive bodies of text all over the application.



And the Forums

Just a few little points about the forums before you start using them.

You'll notice they're very basic, and don't include any sort of like/dislike ability, or even a count of how many people have viewed a post. This is deliberate. You may be wondering why.

In my experience and based on some feedback, the whole like/dislike feature is a great way to compare yourself to others.

"My post only got 10 views and this one got 100! Aw man :("

So, to stop you guys feeling like you may not be important and the like, I haven't added anything like that. Everyone here is equal. My posts aren't more important than yours, and vice versa.

If you tap the help button in the forums section you will get more information about how they work and how to make posts etc, but just remember to be friendly to everyone.

Themes & Donations

Almost done! Just a quick not about the themes in the app (which can be found in the settings menu).

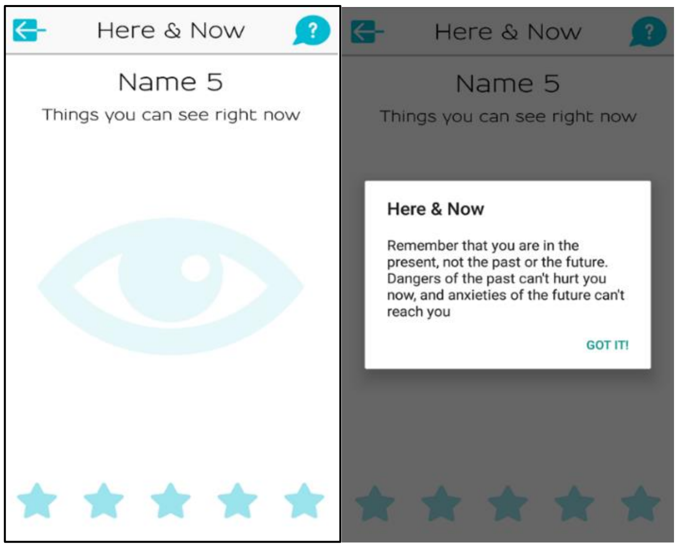
In themes section you'll notice that some of the options are locked and you need to donate to unlock theme. I realise that donating to unlock something sort of defeats the purpose of it being called a donation.

However, I have done my best to make the app free for everyone, meaning you don't need to pay to access content that you might need. As the themes are just for looks and the app is free, I feel it's good to give the themes as a bonus to those who donate towards the app.

Thank you for understanding :)

Figure 8 - What's Up large volume of dialogue

To follow NH of aesthetic and minimalist design, "dialogues should not contain irrelevantly or rarely needed information" [23]. Any added information diminishes the value of viable information. This application is overloaded with text. Instructions contain up to 100 words, demonstrated in Figure 8. This really takes away the relevance of dialogue across the app. Not only are users overloaded with text but it is easy for them to become confused about how to use features of the application. From the outset, there are no instructions about how to go from the introduction pages to the homepage. Users can know where they are in the application from the titles and help icons on every page telling users how to use the application. But there is a huge lack of visibility of system status. For example, on the 'Here & Now' section, there are no instructions or indication on how to complete the task at hand. Experienced users could even find themselves thrown by the lack of guidance. You are required to tap the stars on this activity to move on (Figure 9) but even after consulting the help icon, it takes quite a while to realise to achieve this out.



Here & Now

Remember that you are in the present, not the past or the future. Dangers of the past can't hurt you now, and anxieties of the future can't reach you

Figure 9 - What's Up instruction icon

Despite the busy design of What's Up, the application does have a focus on good usability such as matching the real-world, consistency and error prevention. The vocabulary contained in the application match real-world conventions. It uses everyday terms like "get grounded" to explain overcoming stressful periods. In the same way, the application uses a 'Catastrophe Scale' that allows users to input how bad a scenario was. The scale is based on a slider that converts numbers into emoticons. What's Up speaks the user's language, whether that individual is a novice or expert user, with information provided in a logical order. In addition to real-world conventions, What's Up has a focus on consistency and standards. According to NH, "users should not have to wonder whether different words, situations, or actions mean the same thing" [23]. Throughout the system, images, actions and phrases inevitably mean the same thing. For example, when some individual presses a star icon in the application, it will always have the same spinning action and enlarge. Pressing a star icon always indicates the user has executed the instructions on screen in the application. The conformity within the app minimises for users. Much like having great consistency standards, What's Up puts a spotlight on user prevention. Every page has an exit point through the back arrow in the top left-hand corner. Much like Worry Box and SAM, users can remove any data from the application. As well as that on entering data into the application changes must be committed by answering 'are you sure?' pop-ups. As well as that, users cannot enter data into the application unless all fields are filled in, as all are required to monitor anxiety efficiently.

From What's Up, it can be learnt an anxiety application must speak the user's language and use minimal text to avoid overloading users. The features of What's Up (that help an individual realise their anxiety) are the personal anxiety diary, habit tracker and the catastrophe scale. These features should be implemented in the application.

2.3.3 Conclusion

From evaluating the usability standard and features of the existing solutions, it can be said that the developed system needs to follow some key heuristics. The fundamental heuristics will stop users from becoming further overwhelmed while logging their anxiety. The heuristics are:

1. Speak the user's language by using real-world examples and non-technical terms.
2. Have error prevention for users by including text input constraints and delete action constraints.
3. Avoid users recall terms, icons or actions across the app, but instead invite users to recognise them.
4. Provide users with clear documentation for each activity with minimal text.
5. Have user control and freedom for every action to avoid user's making wrong decisions.
6. Keep users informed throughout the application with labels and titles.

As well as that, the following features must be implemented by the developed system to provide users with an understanding of their anxiety.

1. Anxiety questionnaire/diary
2. List of previous attacks/episodes
3. Anxiety graph/chart
4. Some anxiety rating/scale system

2.4 Other Relevant Research

2.4.1 Introduction

The developed system represents anxiety attacks to users in ways that them to become visually aware of their habits. This representation will come from the information the user has put into the application. For this image to be correct and effective, the user must have a straightforward way of logging anxiety and a simplistic view of the associated trends. The objective of the project is to create an easy to use application. This is achieved uncomplicated design and minimal user input. This will be expressed with the use of emoji inputs and data visualisation. During anxiety attacks, users should have as little interaction with the application as possible. This avoids complicating attacks, meaning as much information as possible about the attack should be automated. This can be done through location and wearables.

2.4.2 Emoji Input

For the developed system to have an easy to use interface and minimal user input, emojis or emoticons could be used to replace text input. Emojis can be described as a "visual representation of an emotion, object or symbol" [24]. and widely used nowadays to represent feelings and actions. According to Dr. Monica Riordan, an assistant professor of psychology at Chatham University, “Emojis help us perform all these little actions.” Emojis, Figure 10, can be viewed as another type of expression for ourselves, much like language [25]. “As long as emoji’s serve a purpose, for communication, self-expression and relationship building we will use them” [25].

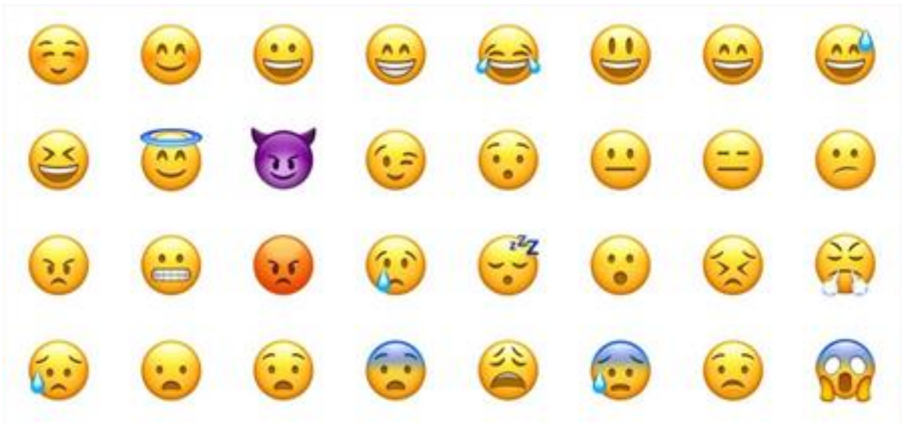


Figure 10 - Emoji’s representing emotion [26]

The first use of emojis or emoticons to express emotion was in 1982. Scott Fahlman, a computer scientist at Carnegie Mellon University, posted the following message to colleagues on a message board:

I propose that the following character sequence for joke markers;

“:-)”

Read it sideways. [27]

Nowadays, Emoji’s need less of an explanation. With the growth of social media and digital communication, these “pictorial representations of feelings” have a significant role in how humans communicate [28]. As the system is intended to be used by all types of users, icons are accompanied by small descriptions to avoid misunderstandings.

The advantage of using icons, such as emojis, in user interface design according to Nielsen Norman Group [29]:

1. Icons are easy targets. For touchscreen platforms, icons can be easily touched.
2. Icons are compact. It is possible to display many icons on a small screen.
3. Icons are recognisable. Standard icons that people have seen and used are fast to recognize.
4. Icons do not need to be translated. Icons generally mean the same for all users.
5. Icons are visually pleasing versus text.

Despite the number of advantages associated with icons, it is important to be mindful of their problems too [29].

1. Icon meanings can change over time. There is no standard for icon meaning. When designing with icons, it is critical to be aware of what icons mean and how likely they are to transform their meaning. Having an icon with an intended meaning that isn't its actual meaning will cause confusion for users.
2. Icons need a label text. This will overcome the issue of icon interpretation. "A text label must be present alongside an icon to clarify its meaning in that particular context." [29]
3. Icons size must aid noticeability. On mobile device screens, icons must compete for attention from users. Users can easily miss content depending on where their attention is drawn to.

2.4.3 Data Visualisation

This application collects information to be presented to users to learn about their anxiety trends. The objective of this application is to allow users to track their anxiety episodes and for counsellors to be able to use that data for the treatment of anxiety. During the interview with Catherine Bolger, she said it was necessary for users to be able to observe their progress in some way that isn't overwhelming and is easy to understand.

Data visualization can be used to turn complex data or huge amounts of information into a visually engaging context. "Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognised easier with data visualisation." This verifies the need for visual representations of anxiety tracking. Humans can process images and charts around sixty thousand times faster than text, as they are "'visually wired' creatures" [30].

This app uses data visualisation to demonstrate the following data to users:

- The situations that cause most anxiety attacks - to allow users to see what is triggering them.
- The correlation between overall mood and leaving a situation - to allow users to realise their reaction to anxiety attacks may not be the best practice for them.
- The location where most anxiety episodes happen - to allow users to map out the environments of their anxiety attacks.
- A timeline of an anxiety attack - to calculate a correlation between stressful life events and attacks

The graphs chosen to represent the data are a pie chart, a line graph and a bar chart. Pie charts are generally used to compare parts of whole data [31]. They don't show any changes over time. A pie chart is suitable to depict the subject causing most anxiety as these are parts of the whole anxiety issue. Whereas line graphs are generally used to compare parts of data that change over periods of time. Correlation is a relationship between quantitative variables [31]. Overall mood and leaving a situation according to Catherine Bolger would have a high correlation and is best measured on a line graph. As previously mentioned, the more a user avoids a situation the more they are diminishing the prospect of overcoming anxiety and as a result leaving their overall mood low. Catherine says it is important to highlight to users that this avoidance behaviour will not benefit them. Finally, bar graphs are generally used to compare values between groups of items [31]. A bar chart will be used to indicate the location where most anxiety episodes happen. A chart will not represent the timeline of anxiety attacks, but instead, the timeline will be represented on a calendar view.

2.4.4 Wearable Technology

The original idea for this project stemmed from the thought that modern wearable technology, such as Fitbit or Apple Watch, could be used to alert users to anxiety attacks i.e. When their heart rate or sweating is over a conventional value during non-workout times. As increased sweating and heart rate are symptoms of anxiety attacks, it was believed the idea could be achieved. From a thread on Fitbit's website [32], a few users expressed the desire for a way to track their heart rate during anxiety episodes with the Fitbit device. The developed system should automate the tracking of anxiety for users as much as possible. This can be using a sensor or wearable to monitor physical symptoms of that attack, paired with a form about an anxiety attack, could be effective in this system.

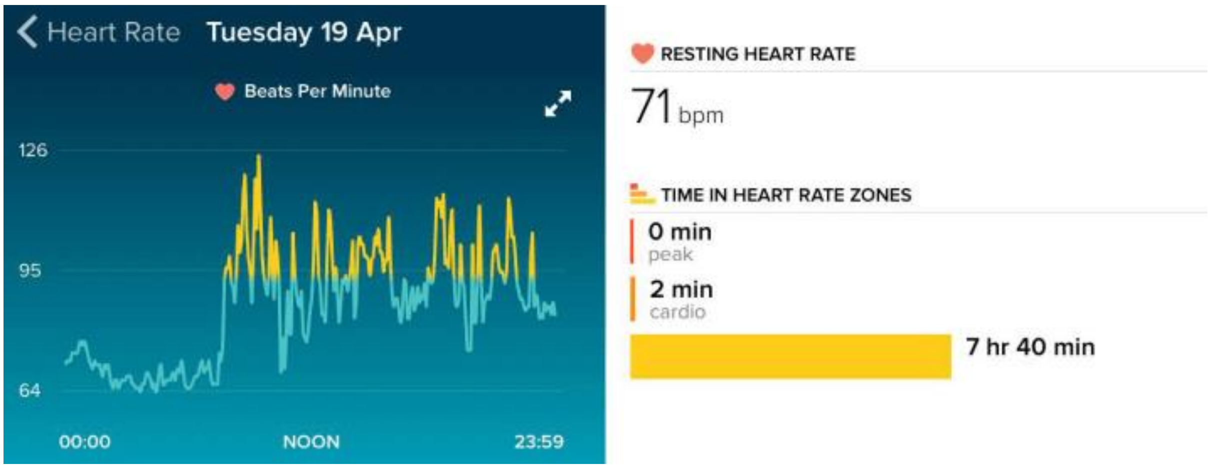


Figure 11 - Fitbit measuring a stress disorder [33]

Figure 11 shows a Fitbit user's beats per minute while sitting at their desk in work, from an article published by Buzzfeed [33]. This user has a stress disorder and it is clear that the Fitbit picks up on stressful periods. Upon visualising their stress, this Fitbit user decided to seek help for their stress as they realised they were doing damage to their body. It is worth noting, that comments from users on the article expressed concerns about seeing massive spikes in heart rate and stated it would cause more anxiety issues. One user said, "I have been diagnosed with anxiety and depression. A heart rate monitor would just play into all my insecurities" [33]. This lead to the decision that the application would not give a live feed of heart rate or sweat rate from sensors. Instead, the developed system alerts users when their symptoms are unordinary and remind them to relax and log their symptoms.

2.4.5 Location tracking

As previously mentioned, the location of an anxiety attack can offer insight into a person's anxiety. The environment of an attack may be an unknown trigger to the individual. “Through regularly tracking panic attacks, you may notice that they often occur when you are in specific situations or events” [11]. Much like using sensors to track physical symptoms, mobile phones can be used to track a user's location. Nowadays smartphones provide GPS services, through the internet or network. To minimise the overall input from users during attacks, the developed system captures the location of the user as soon as a change in symptoms is sensed.

2.4.6 Conclusion

By achieving an uncomplicated design that conveys information visually, users of the system will succeed in tracking their anxiety attacks without being overwhelmed. Minimal inputs and interaction during anxiety attacks will achieve this. The use of emoji inputs, sensors observing individuals and location tracking will allow for non-invasive automation of tracking anxiety attacks. Observing that, data visualisation will bring meaning to track anxiety attacks. From research, it is important that data relating to the user's anxiety attacks and anxiety tracking methods have a positive impact on the users and does not cause any more anxiety.

2.5 Choosing Technologies

2.5.1 Introduction

When the objectives of the project are revisited, it can be noted the developed system should be mobile, easy to use and as non-invasive as possible. From research, the following decisions were made for the developed system:

1. The system should capture changes in user’s physical symptoms during an anxiety attack.
2. The system should capture location at the beginning of an anxiety attack.
3. The system must present anxiety trends to users.
4. The system requires input about attacks from all types of users.
5. The system must provide healthcare professionals and counsellors to access user data.

Technologies need to be researched and chosen based on these decisions. A mobile platform needs to be decided, as well as a sensor that can communicate of that platform. Simultaneously, data needs to flow through the application from the user to a counsellor. This chapter will explore the research undertaken in deciding suitable technologies for the system.

2.5.2 Mobile Technology

2.5.2.1 Introduction

There are numerous amounts of mobile operating systems in use today, with iOS and Android being the most popular and readily available [35]. This section will look at which mobile operating system (iOS or Android) best relates to the objective of the developed system. The operating systems will be evaluated based on availability to users, ease of development including location services and languages.

2.5.2.2 Android

Android is the mobile operating system developed by Google. Android is the “world’s most popular mobile platform”, powering “hundreds of millions of mobile devices in more than 190 countries around the world” [36]. The objective of this project is to develop a system that can be used by users of all levels of IT skills. Android is generally more widely used by people due to its low-cost and simplified user interface [37]. “With the Android operating system, customers have access to a range of different smartphones that are all able to run Android, which may explain why it has a higher loyalty rate” [37]. This is an advantage of using Android for the developed system as it the platform that could reach more users.

Location services on Android are easily achievable. “The location APIs available in Google Play services facilitate adding location awareness to your app with automated location tracking, geofencing, and activity recognition” [38]. Using Google Play service's location APIs, an android application can request the last known location of a user’s device. This location is predominantly the user’s current position. Android applications that require location knowledge about the device must request location permission from users. Despite requiring permission all other development for getting location is straightforward.

As well as that, from college modules and projects there is a prior knowledge about developing Android systems. The main language associated with Android is Java, which has a huge number of open-source libraries that may be useful during development given the time risk. Additionally, an Android device is readily available for testing and development. This makes Android a suitable platform for this project.

2.5.2.3 iOS

iOS is the mobile operating system developed by Apple Inc. Unlike Android, iOS is fixed to mobile devices manufactured by Apple. Meaning users are limited to only one type of device. iOS may be a good contender for the developed system as apps developed on it are usually very simple and crisp [39]. This allows users to focus on what they need to do at a particular time. As well as that, iOS devices are becoming common productivity tools in business. “Doctors and nurses at hospitals are beginning to use iPads to view X-rays and CT scans and read medical records while standing next to the patient” [39]. This relates to counsellors and healthcare professionals using the developed system to understand their patient's anxiety. Despite the use of iOS in hospitals and workplaces, the focus of this system lies with users with anxiety. Similarly, there is a high cost associated with iOS devices and a lack of an assortment. This could be a possible disadvantage of developing with iOS as the variety of devices is very small.

Location services on iOS appear similar to that of Android. “The Core Location framework lets you locate the current position of the device and use that information in your app” [40]. There are two services used to get users location. Standard location service and significant-change location. The disadvantage of using iOS to gather a user’s location is that it can be power-intensive. “For most apps, it’s usually sufficient to establish an initial position fix and then acquire updates only periodically after that” [40]. As anxiety attacks can happen at any time, the application must be up to date with the user's location.

iOS applications are built using Swift and Objective-C. This is not a familiar language and requires XCode. XCode is the development platform for iOS only available for MacOS. MacOS is not a readily available platform for development, nor is an iOS device for testing. This makes iOS not an attractive operating system for this project.

2.5.2.4 Conclusion

In conclusion, the Android operating system is the best fit for the development of this application. Android development is a previously known operating system with a great range of libraries and location services.

2.5.3 Sensors

2.5.3.1 Introduction

The proposed system will let users track the physical symptoms of anxiety. As previously outlined, the physical symptoms of anxiety include an unusual heartbeat and breathing, feelings of warmth, sweating, tension in muscles, nausea, headaches, fatigue, crying spells, and change in appetite. Following initial research, it is true that only a subset of these symptoms can be monitored by existing sensors. These physical symptoms include unusual breathing, unusual heartbeats and sweating. This section will evaluate the suitability of different sensors for this application with a developed evaluation criterion. The sensors examined are galvanic skin response sensor, pulse sensor, a band belt respiration sensor and a moisture sensor.

2.5.3.2 Evaluation Criteria

To evaluate the suitability of sensors for the developed application, a criterion was developed. The following standards need to be met for the sensor to be appropriate (in alliance with the objective).

- The sensor must be easily attainable. This includes reasonable delivery wait, purchase cost, and documentation.
- The sensor must be integrable. The chosen mobile platform is Android, any chosen sensor needs to serve this. (including a usable format of incoming data)
- The sensor must be flexible. There needs to be an allowance for trial and error.
- The sensor must be easy to use and non-intrusive as possible.

It is vital to note that as this project is a proof of concept and sensors used during development may not be as non-invasive as a developed product for a market. For this developed system, a non-invasive sensor is one that does not cause more anxiety for a user and is mobile.

2.5.3.3 Galvanic Response Sensor

Galvanic Skin Response Sensor (GSR) is a sensor that measures the electrical conductance of skin. Emotion causes a stimulus to the nervous system, resulting in physical changes to the body such as sweating. Grove – GSR sensor recognizes strong emotional changes (as a change in resistance) by attaching electrodes to two fingers on one hand [41].

Grove GSR sensor is an easily attainable device with a middle range cost. There is access to documentation for this sensor online, as well as compatibility with Raspberry Pi and Arduino [41]. There are many set up tutorials online for connecting the sensors to an Arduino or Raspberry Pi. As it is suitable for Arduino and Raspberry Pi, the input data is a converted voltage to a digital value. Grove GSR sensor is worn on two fingers, this is non-invasive [41]. The sensor is highly credible. It is used in many successful projects documented online.

2.5.3.4. Pulse Sensor

The pulse sensor is a sensor that measures the volume of blood being pumped to extremities. Changes in blood volume can be detected by shining a light through the finger and calculating the amount of light that can be passed through it. This process is used with a conductor called a photodiode [42].

A pulse sensor is easily available to purchase that is inexpensive. There is access to adequate documentation for this sensor online. The pulse sensor is compatible with Raspberry Pi and

Arduino. There is documentation available at pulsesensor.com for set up. As it is suitable for Arduino and Raspberry Pi, the input data is a converted voltage to a digital value. The pulse sensor is worn on one finger, making it a non-invasive sensor. Many projects use the sensor to measure heart rate and heartbeats.

2.5.3.5 Band Belt Respiration Sensor

Band Belt Respiration Sensor is a sensor that measures the rate in which a person's chest is moving in and out, to get oxygen in. The sensor is a belt that wraps around a person's chest and when the chest expands and contracts, the resistance of the belt changes. [43] Based on the rate of this changing, it can be detected whether respiration rate is normal or not. "The normal respiration rate for an adult at rest is 12 to 20 breaths per minute." [44]

A band belt respiration sensor is not easily available. This sensor must be built prior to set up. There is some documentation for the development of the sensor, but not much. The cost of this sensor would be excessive and some components needed would be difficult to obtain. A band belt respiration sensor can be made compatible with Raspberry Pi or Arduino. There would be an immense amount of time and effort involved to set up this sensor, without access to documentation.

2.5.3.6 Moisture Sensor

A moisture sensor is a sensor that measures the "dielectric permittivity of the surrounding medium" [45]. This essentially indicates the amount of water that is flowing through the surrounding medium. The sensor creates a small "voltage proportional to the dielectric permittivity, and therefore the water content" [45]. The cheapest sensor of this kind would be a soil moisture sensor.

A soil moisture sensor is an easily attainable device with a low cost. There is access to documentation for this sensor online, although there is little to no documentation of projects involving measure moisture on a human's skin. Soil moisture sensors that are compatible with Raspberry Pi and Arduino are available. There are many set up tutorials online for connecting the device to an Arduino or Raspberry Pi. As it is suitable for Arduino and Raspberry Pi, the input is a converted voltage to a digital value. Some restrictions may apply when trying to measure moisture on a human's skin. A soil moisture sensor would need to be worn in an area prone to sweat. I.e. under armpit area or palms of the hand. This is a non-invasive detector. Because of the voltage being passed from the sensor, this may not be a suitable sensor for humans.

2.5.3.6 Conclusion

The following table summarises the evaluation of the researched sensors.

	Grove - GSR	Pulse Sensor	Band Belt Sensor	Moisture Sensor
Attainable	Yes	Yes	No	Yes
Integrable	Yes	Yes	No	Yes
Flexible	Yes	Yes	Yes	No
Usable	Yes	Yes	No	No

From the evaluation, for this proposed system Grove - GSR sensor and pulse sensor are suitable reliable sensors. This project promises a non-invasive easy way for users of all levels to track their anxiety subconsciously, which can be achieved with the GSR sensor and pulse sensor. Due to the risks of sensors failing during development and testing, both sensors were purchased and set up.

2.5.4 Mobile and Sensor connection

2.5.4.1 Introduction

The sensors chosen for the developed system cannot be paired directly with an Android device. The data coming from the sensor needs to be converted from the input voltage to digital values by another device before the android device can receive updates. The imagined architecture for this relationship is viewed in Figure 12. This can be done using a microcontroller such as Arduino or Raspberry Pi. These microcontrollers are supported by the chosen sensors. This section will consider Arduino and Raspberry Pi as sufficient candidates for the developed system.

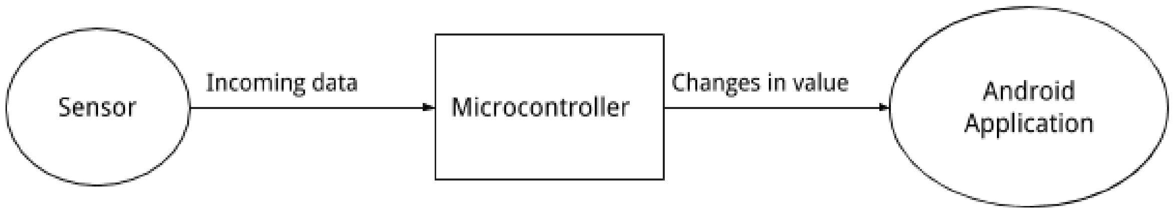


Figure 12 - Connection between sensor and application

2.5.4.2 Arduino

“Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs such as light on a sensor” [39]. The microcontroller board can be programmed and be given specific instructions. Arduino is a simple computer that can execute those instructions (as a program) repeatedly. Arduino Uno is a type of Arduino board suitable for “simple repetitive tasks” [46] (including the primary task of this project). Arduino is also known to be the best microcontroller for beginners which makes it attractive for this project.

2.5.4.3 Raspberry Pi

A Raspberry Pi is a small computer that has the ability to run multiple programs at once. It is suitable for intensive computations. It is more complicated than an Arduino [46]. If a project requires two tasks to be carried out at the same time then a Raspberry Pi is suitable. For this system, the sheer power of a Raspberry Pi is unneeded.

2.5.4.4 Conclusion

For the developed system, an Arduino is the microcontroller of choice. One of the intended uses of an Arduino is reading values from sensors. It is also an easy microcontroller to set up and use as a beginner

2.5.5 Database

2.5.5.1 Introduction

The developed system allows users and counsellors to have access to the same data about the user's anxiety attack. As well as that, the project objective states that the system developed will be a mobile system. This means the data contained and inputted to the application will need to be accessible on multiple devices. The following section will discuss the evaluation of local Android databases and cloud-based databases.

2.5.5.2 Local Storage

SQLite is the default open source SQL database of Android. The database stores data on the Android device, this means that the database is local to a given device [47]. The advantages of using the local database on the mobile device include the database always being available, data retrieval from an SQLite database being more robust and data accessed using familiar SQL queries [48].

Despite the advantages of SQLite, a local database could be unused for all data storage associated with this developed system. For counsellors/healthcare professionals to be able to see updates on their client's anxiety, data needs to be stored and retrieved remotely.

2.5.5.3 Remote Storage

Firebase is a remote real-time database that uses NoSQL. Firebase is built on the same infrastructure as Android (Google) [49]. This makes it a competent candidate for remote storage of the data in this system. Firebase can store and retrieve data the user inputted into the application, even when there is no network connection. Firebase automatically pushes new data to the Firebase NoSQL database when a network connection is gained [49]. This means the application is available to the user to store anxiety episodes offline. Firebase is good for prototyping an application and more limited applications [50]. Firebase is easily integrated with Android and has an abundance of documentation available, making it a good contender for storage for the developed application.

2.5.5.4 Conclusion

In conclusion, Firebase is the main storage for this application. After some development work, it was realised that a combination of the local SQLite and Firebase database is an adequate solution. SQLite is used to store stock response to anxiety attacks within the application and Firebase is used to store mostly all other data.

2.4.6 Overall Conclusion

Android fulfils the objective of the project because of the vast population of Android users, as well as previous experience with development of Android applications. Java is the main development language of the developed system. It is suitable for the project as it integrates the chosen storage SQLite and Firebase smoothly, and has many open-source libraries that can be utilised to avoid time issues. Because of the perceived risks with sensors, two were chosen: galvanic skin response and pulse sensors. These devices are easy to use, available and flexible. The decision was made to communicate with the Android device, and sensors would be carried out by an Arduino Uno. This microcontroller is suitable for smaller projects and manageable for beginners.

2.6 Resultant Findings and Requirements

From the research gathered, the requirements for the developed system can be outlined. This section will consider the research acquired and how it directly relates to the requirements of the system.

2.6.1 Resultant Findings

Firstly, with the gained understanding of anxiety, how it manifests, who it affects, and how it can be tracked, an input for logging attacks can be designed. Anxiety has a very broad spectrum of whom it affects and how it affects people, meaning any developed anxiety application must reflect this. When tracking anxiety through the developed system subject, thoughts, physical symptoms and behaviours, mood, reaction and location must all be taken note of. Sweat and heart rate are physical symptoms of anxiety that can be automatically tracked by body sensors.

An anxiety tracking application must observe the practices of effective anxiety tracking by presenting data to users visually. This means inputs should be minimal and use icons where possible. As well as that, trends need to be obtained from the data inputted by users and presented in charts. From acquiring an understanding of personal anxiety manifestation, users in conjunction with healthcare professionals can begin treatment for individuals. Giving counsellors access to tracking applications used by their clients, it provides them with constant feedback about their client's anxiety trends. Which is why the developed system will ensure counsellors have adequate access to data about their clients.

Alternative existing solutions demonstrate the features desired by users and the correct design standards that need to be followed when developing the system. Any mental health applications must be easy to use and involve little user input. The applications must follow Nielsen's Heuristics for User Interface Design. The Worry Box, SAM and What's Up validated that the application needs to speak the user's language especially in an application looking to tackle mental health issues. Similarly, error prevention needs to be at the forefront of the developed system along with clear documentation for each activity. The overall goal of this project includes maintaining an easy to use application for users will all IT levels.

Due to the broad types of user for this system, the technology of the system must be usable by everyone. The mobile platform chosen is Android, the most popular operating system meaning any application developed for it will be reachable by many people. Any sensors connected to this system must also be suitable for any type of user. The chosen non-invasive Grove – GSR sensor and Pulse sensor simply connect to a user's finger and begin sensing their anxiety episodes.

From these findings, the user requirements can be created for the system.

2.6.2 User Requirements

The user requirements for the developed system are as follow:

- As a regular user, I need to be able to input my anxiety attacks into the application via questionnaire or wearing sensors
- As a regular user, I need my location captured by the application when I have an anxiety attack
- As a regular user, I need as little interaction with the application when I am feeling anxious
- As a regular user, I need to be given a visual representation of my anxiety trends, such as my mood, most anxious location and common subject
- As a regular user, when I have an anxiety attack I need to be notified that the app is tracking me and that I need to log the attack
- As a regular user, I need a timeline of my previous anxiety attacks
- As a regular user, I need to feel my data is safe and protected with a login and only registered counsellors can see my anxiety profile
- As a regular user, I need to be able to access my account from any Android device and not fear losing my data if I lose my phone
- As a counsellor user, I need to have access to my patient's profile

From the user requirements, the features of the application can be formed.

Chapter 3 Design

3.1 Introduction

Given the resultant findings and established user requirements, a design must be created for the proposed system. The design phase must include defining the front-end and the back-end of the system. Designs include use case diagrams, technical architecture design for the system, storage design and code design. These designs will aid the development of the system and the creation of prototypes for the system. A methodology for the project needs to be decided before any designing or developing can begin.

3.2 Methodology

3.2.1 Introduction

This project is one that is carried out by a single individual, that requires user testing and input to be successful and follows a time limit. The chosen approach must reflect the needs of the project. The plan must be flexible, i.e. be able to revisit features if users are unhappy with them. The method must include giving priority to sections over others (because of time restrictions and risks) and, adopt a team-based methodology for a single person. This section will explore the suitability of the Waterfall and Agile Methodologies for this project and the reasons for the chosen methodology.

3.2.2 Choosing a methodology

Waterfall Methodology

The Waterfall Methodology is defined as a linear and sequential approach to software development, Figure 13. There are eight stages involved in development using Waterfall [52]:

- Gather and document requirements
- Design
- Code and unit test
- Perform system testing
- Perform user acceptance testing (UAT)
- Fix any issues
- Deliver the finished product

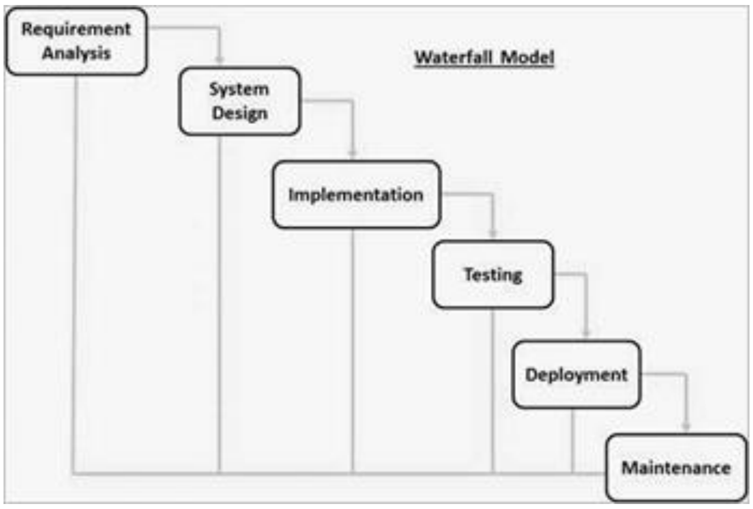


Figure 13 - Waterfall project lifecycle methodology [53]

Each stage is completed fully before moving on to the next stage. Because of the linear and sequential properties of Waterfall, any stage that has already been completed cannot be revisited. To return to any completed stage would result in the project being started all over again. “There’s no room for change or error” [52].

How the Waterfall methodology is suitable for this project [52]:

- With waterfall methodology, the end system is known and expected. There will be an idea of size, cost and a timeline for the project. There will be a rigid expectancy of what the system will do at the end of the project.
- Progress will be easily measurable, as the full scope of the project will be known in advance. Which appeals to the time restraint of this project.
- The design of the system will exist prior to development and will never change thereafter.

How the Waterfall methodology is unsuitable for this project [52]:

- If the initial requirements of the project are not solid (i.e. missing any amount of information), the project will be unsuccessful.
- The delivered system may be dissatisfactory, and by the end of any changes will be costly.
- Once one stage of the project is complete, it can never be returned to without restarting the whole project.
- As the system will only be tested at the end of development, any bugs that arise may have a huge effect on existing code.

The Waterfall Methodology is unsuitable for the design and development of this system. This methodology interprets the design and scope of the project in advance, with no change. That is too severe for this project, there needs to be room for error and change. As well as that, leaving all testing to the end of development is too risky for this project.

Agile Methodology

The Agile methodology is an incremental and iterative approach to software development, Figure 14. Thanks to its iterative approach, Agile development allows developers to be flexible and respond to any critical errors that arise during any time of the project [52]. Developers and designers, as part of a team, can be flexible to change requirements and revisit any stages.

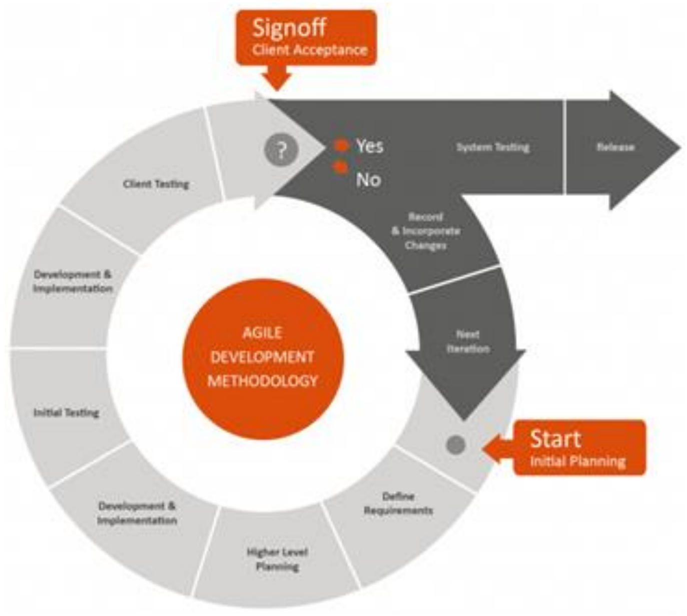


Figure 14 - Agile Project Lifecycle Methodology [54]

How the Agile methodology is suitable for this project [52]:

- Changes can be made throughout any stage of development and almost anticipated.
- Testing occurs at the end of every sprint. Bugs found can be taken care during the development cycle and doesn't add to the cost at the end.
- Frequent work can be delivered and any changes can be made if the work is unsatisfactory.
- With time constraints, a full set of features does not have to be completed and a basic version of working software can be released.
- Development is more user-focused and direction from users can be included at any stage of development.

How the Agile methodology is unsuitable for this project [52]:

- Agile development requires complete dedication to the project due to the frequent changing of requirements
- Due to time-bound deliverables, some features cannot be completed within a sprint and additional unplanned sprints may need to utilised. This adds to the project cost.
- Without an initial solid plan, the end system can be hugely different to the imagined.

The Agile Methodology is a suitable methodology for this project. Due to the incremental and iterative nature of the methodology, changes can be easily made to design and scope. Due to the time constraint of the project, a basic version of operating software can be developed under this methodology. As well as that, the Agile methodology is closely associated with the goal of having a user-focused project. With time constraints, a full set of features does not have to be completed and a basic version of working software can be released.

3.3.3 Chosen Methodology

The chosen methodology that was used to design and develop the system was Agile following scrum framework. Given the time constraint of the project, Scrum development will work adequately. Scrum development generally works in sprints of around 10-14 days.

For this project, all designing and user requirement defining will be done up front followed by development sprints. Generally, in industry, design teams work two sprints ahead of development teams to ensure designs are complete before development. Although the Agile is flexible, for this project this flexibility cannot be fully utilised without risk of not completing the project. To ensure the system is robust, design and user requirements need to be fool-proof before any code is produced to avoid risks in affecting any existing features.

The research and design phase will include a huge amount of research using the design thinking process. “Design Thinking is a method used by designers to solve complex problems and find desirable solutions for clients. A design mindset is not problem-focused, it’s solution focused and action-oriented towards creating a preferred future.” [53] The application will have a user centred design.

Following an extensive amount of research, an overview of the system and user requirements will be defined from this research. A high-level design of the system will be created. Once requirements are defined and agreed on they will be unchanged and a feature list can be developed. From the user requirements and high-level scope, prototypes of the system will be created, to guide technical design and development.

Once research and design have taken place, then the development sprints will begin and will have the following cycle like that in industry:

- Follow defined designs (from user requirements)
- Develop the feature
- Test the feature
- Fix any remaining bugs

These steps must be completed before moving on to the development of the new feature. This will ensure feature completeness, meaning features with a higher priority will be completed fully and placed in the system.

The approach is flexible and allows iteration on certain features of the project. The approach gives priority to sections and allows users to be involved in testing and planning. Having chosen a life cycle method, the design and development of the project can begin. A design of each component of the system can now be produced.

3.3 Feature List

To incorporate the user requirements into the developed system, the following features were decided upon:

- *Anxiety attacks questionnaire*
  - Users log their anxiety attack symptoms in a questionnaire. There are asked to input the subject of anxiety, their thoughts, physical feelings, moods, the rating of their mood and the reaction to the anxiety attack. The data captured by the questionnaire will be the base of the insights into the user's anxiety.

- *Anxiety attack list or "sensed anxiety"*
  - Users can view all past anxiety attacks felt by sensors or submitted through a questionnaire in a list activity. From this list, they will be able to view any questionnaires that have been logged for that sensed anxiety.

- *Calendar of anxiety*
  - Users can view all past anxiety felt by sensors or submitted through a questionnaire in a month calendar view. From the calendar, they will be able to view any questionnaires that have been logged for a particular date.

- *Anxiety trends*
  - Users can view the trends in their anxiety. They can see how location, reaction and subject influences their anxiety. The anxiety trends will be presented as charts.

- *User Login*
  - Once a user is registered, users can log in to the application. During registration, a user will decide if they need a regular profile or a counsellor profile.

- *Reminder to tracks*
    - Users receive a notification once a day to remind them to track their anxiety. Future work could include personalising the notification to match times when users are prone to having anxiety attacks.

- *Alerts during anxiety episodes*
    - Users receive a notification to log their anxiety attack when sensors feel some change in physical symptoms.

- *User profile*
    - Users have a profile where they can allow counsellors have access to their anxiety.

- *Counsellor profile*
    - Users who are counsellors have a way to choose which patients anxiety they are viewing.

- *Sensing anxiety*
    - Users will wear a device to monitor their physical symptoms of anxiety. When changes are sensed they are invited to log their anxiety attack.

3.4 Use Case Diagram

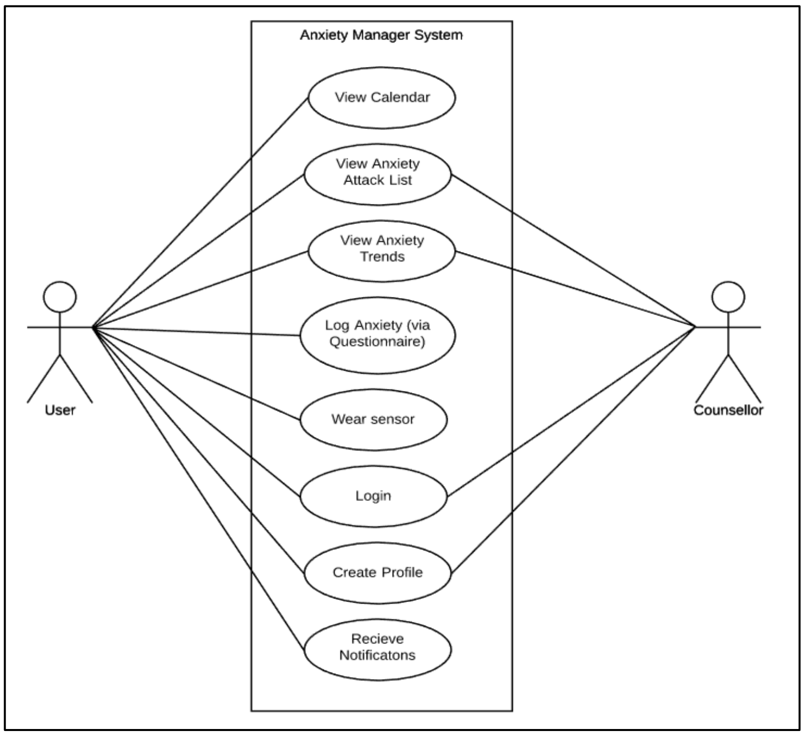


Figure 15 – Use Case diagram for Anxiety Manager system

The use case diagram for the proposed system is in Figure 15. The system consists of seven use cases and two actors. In this system, a regular user can create an account and then log in. The user can view their calendar, their sensed anxiety and their anxiety cycle. They can log their anxiety by filling out the questionnaire. They can also receive notifications when an anxiety episode is sensed. In this system, a counsellor user can create an account and log in as well as a regular user. They can only view a user's sensed anxiety and anxiety cycle.

3.5 User Interface Design

3.5.1 Introduction

This section outlines how the final user interface design was decided. A few horizontal interactive prototypes were created from the user requirements. Users were asked to give feedback on horizontal prototypes. This led to a couple of iterations of the design until all requirements and standards previously developed were followed.

3.5.2 First Iteration

The first iteration of the design phase was to use pen and paper to draw the user requirements roughly to mock up the application, shown in Figure 16.

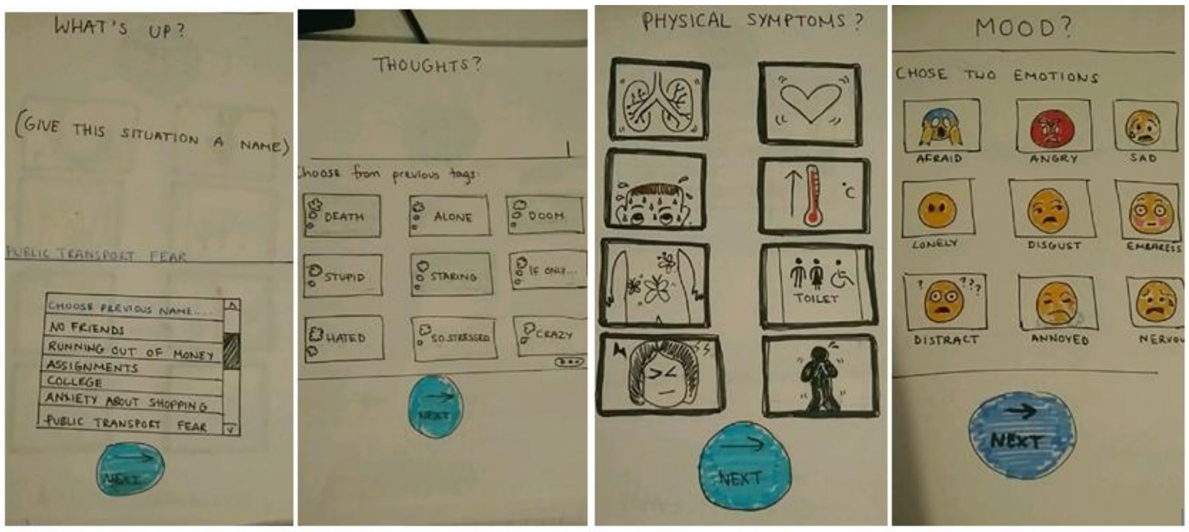


Figure 16 - Sketched mock-ups of the application questionnaire

After hand drawing the application, a prototyping application was used to envision the interaction between features. The application was called 'POP.' "POP helps you transform your pen and paper ideas into an interactive iPhone or Android prototype" [54]. The POP app allows users to take pictures of their drawings and put hotspots on certain parts of their paper mock-ups. These hotspots allow the paper drawings to be interactive, Figure 17.

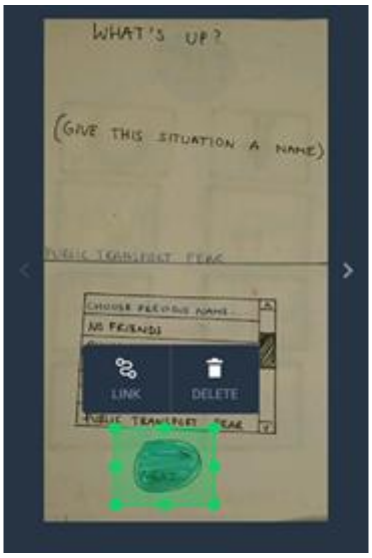


Figure 17 - Adding hotspot to paper mock-ups

Once the interactive hand-drawn prototypes were completed, a link to them was sent to users. The users were of varied IT levels and ages: user one, a twenty-one-year-old with a high level of IT skills and user two, a forty-five-year-old with a low level of IT skills. The feedback given included:

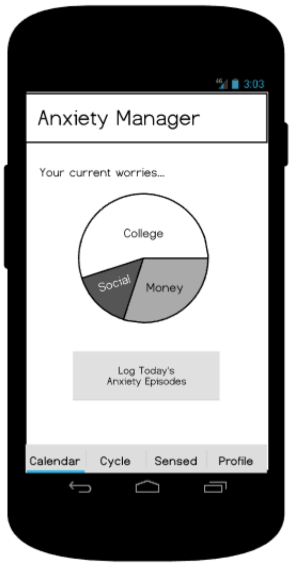
"This is really good. It is easy to understand the flow of the app, and the icons are good too. I would move the next button to somewhere else. It would also be better if you moved on to the next question when you selected your answer. If I'm logging anxiety after an attack, I want it to be as quick as possible." - User one

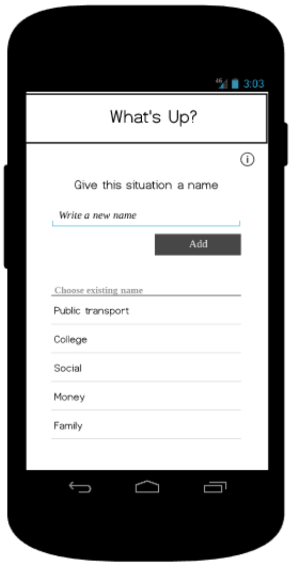
"Sometimes it was hard to understand what to do next. I was looking for instructions on what to do but I couldn't find them like other apps I use. The pictures were nice and meant I didn't have to read." - User two

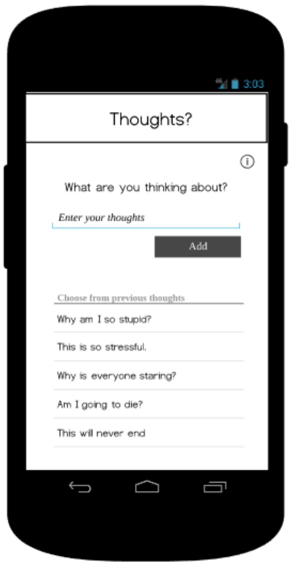
Following the feedback from users, the next iteration of the design phase could begin.

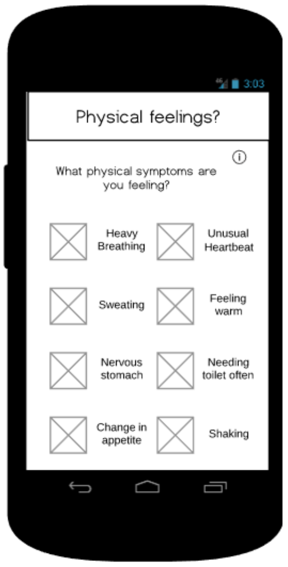
3.5.3 Second Iteration

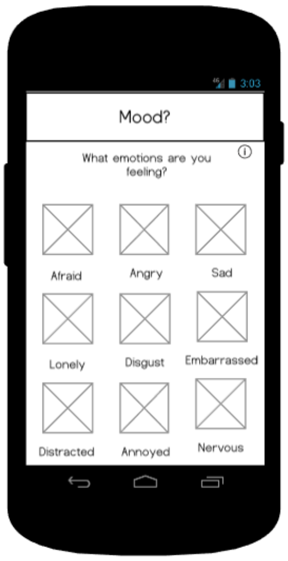
From the feedback received from users, a new version of the prototype was created using 'Lucidchart' instead of pen and paper. Lucidchart has a palette of mobile icons that makes creating wireframes easy. The following wireframes were created.

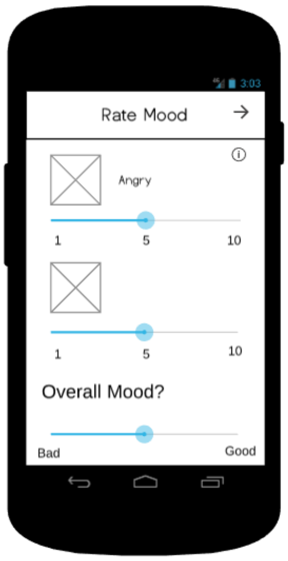

3:03
Anxiety Manager
Your current worries...
College
Social
Money
Log Today's Anxiety Episodes
Calendar
Cycle
Sensed
Profile

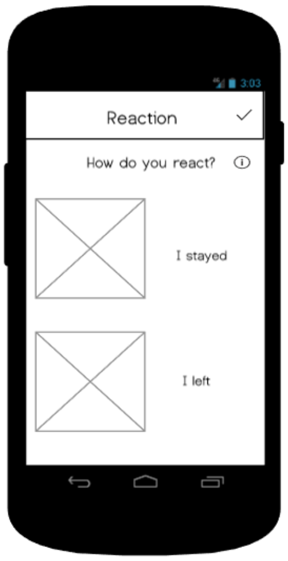

3:03
What's Up?
Give this situation a name
Write a new name
Add
Choose existing name
Public transport
College
Social
Money
Family

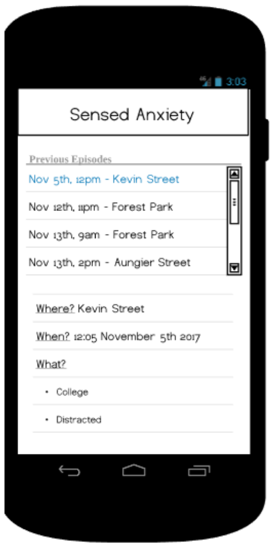

3:03
Thoughts?
What are you thinking about?
Enter your thoughts
Add
Choose from previous thoughts
Why am I so stupid?
This is so stressful.
Why is everyone staring?
Am I going to die?
This will never end

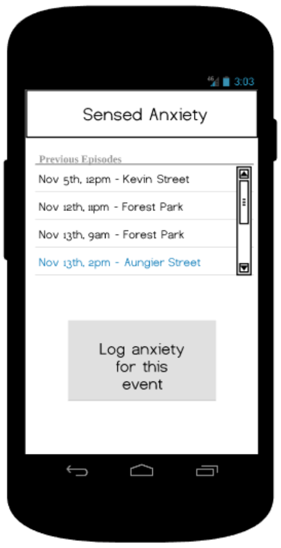

3:03
Physical feelings?
What physical symptoms are you feeling?
Heavy Breathing
Unusual Heartbeat
Sweating
Feeling warm
Nervous stomach
Needing toilet often
Change in appetite
Shaking

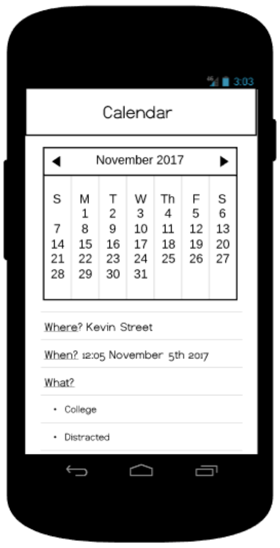

3:03
Mood?
What emotions are you feeling?
Afraid
Angry
Sad
Lonely
Disgust
Embarrassed
Distracted
Annoyed
Nervous

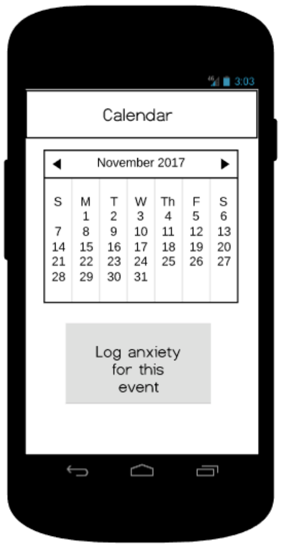

3:03
Rate Mood
Angry
1
5
10
1
5
10
Overall Mood?
Bad
Good

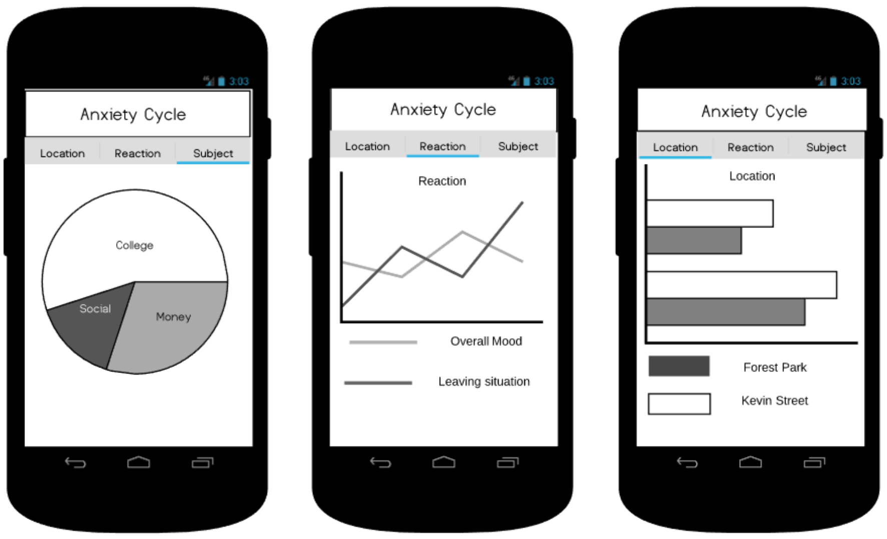

3:03
Reaction
How do you react?
I stayed
I left

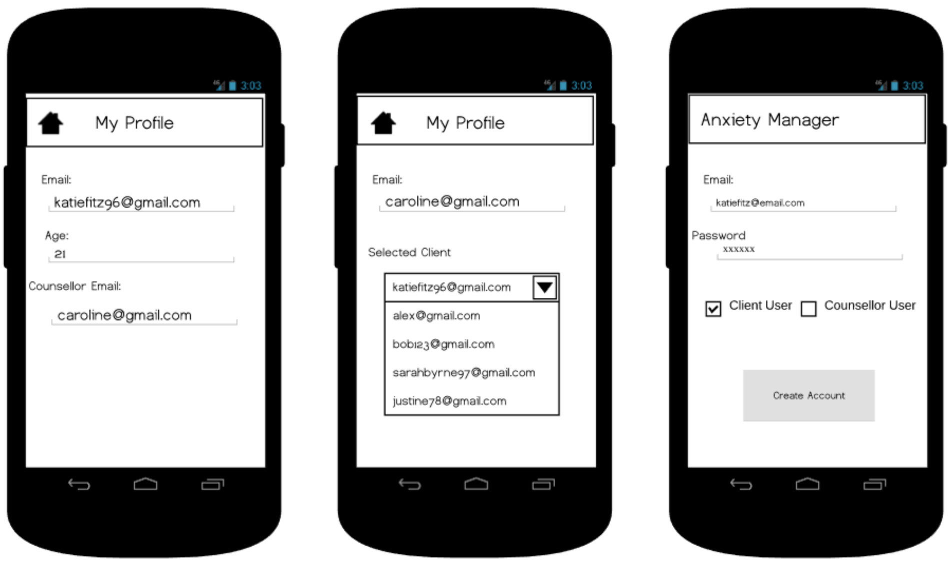

3:03
Sensed Anxiety
Previous Episodes
Nov 5th, 12pm - Kevin Street
Nov 12th, 11pm - Forest Park
Nov 13th, 9am - Forest Park
Nov 13th, 2pm - Aungier Street
Where? Kevin Street
When? 12:05 November 5th 2017
What?
College
Distracted


3:03
Sensed Anxiety
Previous Episodes
Nov 5th, 12pm - Kevin Street
Nov 12th, 11pm - Forest Park
Nov 13th, 9am - Forest Park
Nov 13th, 2pm - Aungier Street
Log anxiety for this event


3:03
Calendar
November 2017
S M T W Th F S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
Where? Kevin Street
When? 12:05 November 5th 2017
What?
College
Distracted


3:03
Calendar
November 2017
S M T W Th F S
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
Log anxiety for this event





Email: katiefitz96@gmail.com
Age: 21
Counsellor Email: caroline@gmail.com

Email: caroline@gmail.com
Selected Client: katiefitz96@gmail.com
alex@gmail.com
bob123@gmail.com
sarahbyrne97@gmail.com
justine78@gmail.com

Email: katiefitz@email.com
Password: xxxxxx
[✓ Client User] Counsellor User

Then using POP again, hotspots were placed over the wireframes and made interactive. Again, the prototypes were sent to the same users via a link and feedback was received. The feedback given the second round included:

"This was much better. It is even easier to understand the flow of the app, although the icons were missing I had a fair idea of what it would look like. Having the questionnaire move with my answers was what I wanted." - User one

"It was easier to understand what to do this time around. There were icons in the questions section which I guess gives instructions. Missing icons though. Having the calendar is useful cause that's how I track panic attacks at home." - User two

The second iteration of the horizontal prototype was the decided prototype for the application interface, although when development became some minor changes were made to accommodate time challenges.

3.5.4 Conclusion

The horizontal prototyping of this application with hand-drawn mock-ups was successful and allowed users to be involved by giving feedback on the prototypes. This feedback supported an iteration of prototype creation to take place and from there a final design could be decided on.

Based on that design, a technical architecture design could be established.

3.6 Technical Architecture Design

3.6.1 Introduction

The architecture for the developed system of this project is a standard three-tier model, exhibited in Figure 18. This model was chosen as any changes made to one tier do not affect any other tiers. The first tier is the presentation tier. This tier includes all the interaction a user has with the system. The second tier is the middle tier. This is the tier that contains the logic. The third tier is the data tier, that contains the data of the application. This section gives a high-level understanding of the applications architecture.

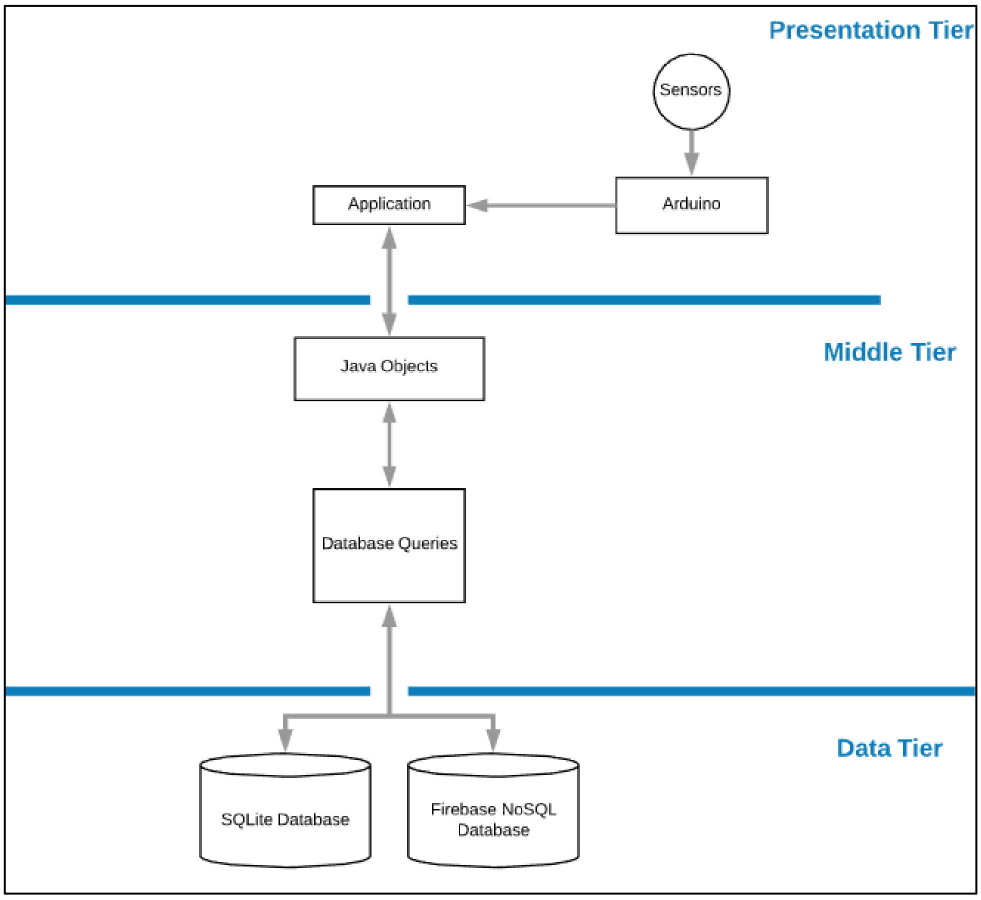


Figure 18 - Technical architecture diagram

3.6.2 Presentation Tier

The first tier is the presentation tier. This tier includes all the user interface components of this proposed application and the sensors that will be worn by users.

App Interface and Sensors

This is the top level of the proposed system's presentation tier. This is the tier where the user of the system interacts with the application components (mobile phone and sensors). Here the user will be using sensors that will be tracking their physical symptoms of anxiety and then filling in questionnaires based on those episodes. The presentation tier will collect data from an instantiated object and display it to the user (e.g. questionnaire inputs). This tier is only for presenting the data to the user.

Arduino

This is the second most top tier of the proposed system. This is the component that interacts with the sensor and the application. Here the data from the sensors and the user's physical symptoms will be collected by the Arduino, any changes in physical symptoms will cause a unique value to be sent to the application.

3.6.3 Middle Tier

Java Objects

The Java object provides access to databases. This object performs all database interactions. The Java objects will be used to retrieve data from the database through Firebase scripts and parse the data to be used in the presentation layer. The objects also push data to the database from interactions on the mobile phone or sensors. Here data will be parsed to JSON for use in the local and remote databases.

Firebase Scripts

The server layer of this architecture uses Firebase to query the Firebase NoSQL database. For example, when there is an attempt to log in, the application use Firebase code query the database to check the username and password are correct. The Firebase database will send back a NoSQL JSON object indicating if the login was successful or not to the application. This information will then be sent to the presentation tier and displayed to the user through the mobile application.

3.6.4 Data Tier

Firebase Real-time Database

The remote Firebase Real-time Database is NoSQL. The Firebase Real-time Database "stores and syncs data with our NoSQL cloud database. Data is synced across all clients in real time, and remains available when your app goes offline". For this proposed system, Firebase will store and retrieve data the user inputted into the application into the NoSQL database, even when there is no network connection. Firebase automatically pushes new data to the Firebase NoSQL database when a network connection is gained. This means the application will be available to the user to store anxiety episodes offline.

SQLite Database

The local Android database is SQLite. For this proposed system, SQLite will store static information needed by the application to populate list views and charts for example.

3.7 Storage Design

3.7.1 Introduction

From the determined features of the system, the following data needs to be stored for the system to be usable. During the questionnaire, the subject, thought, physical symptom, emotions, the rating of mood, and reaction needs to be stored. A unique id needs to be created for every questionnaire. This data will be stored remotely. During an anxiety attack, the location, current timestamp needs to be stored. A unique id also must be created for every anxiety attack. During registering, the username, email, and password need to be stored along with a value indicating whether a user is a counsellor or not. A unique id needs to be generated for every user. This section will explore the iteration undertaken to determine the design for storage.

3.7.2 First Iteration

The first iteration of the database design did not consider two types of storage and only focused on cloud storage. The first design consisted of seven tables. Although the remote database is NoSQL database an entity-relational diagram is used to demonstrate relationships between the data, shown in Figure 19.

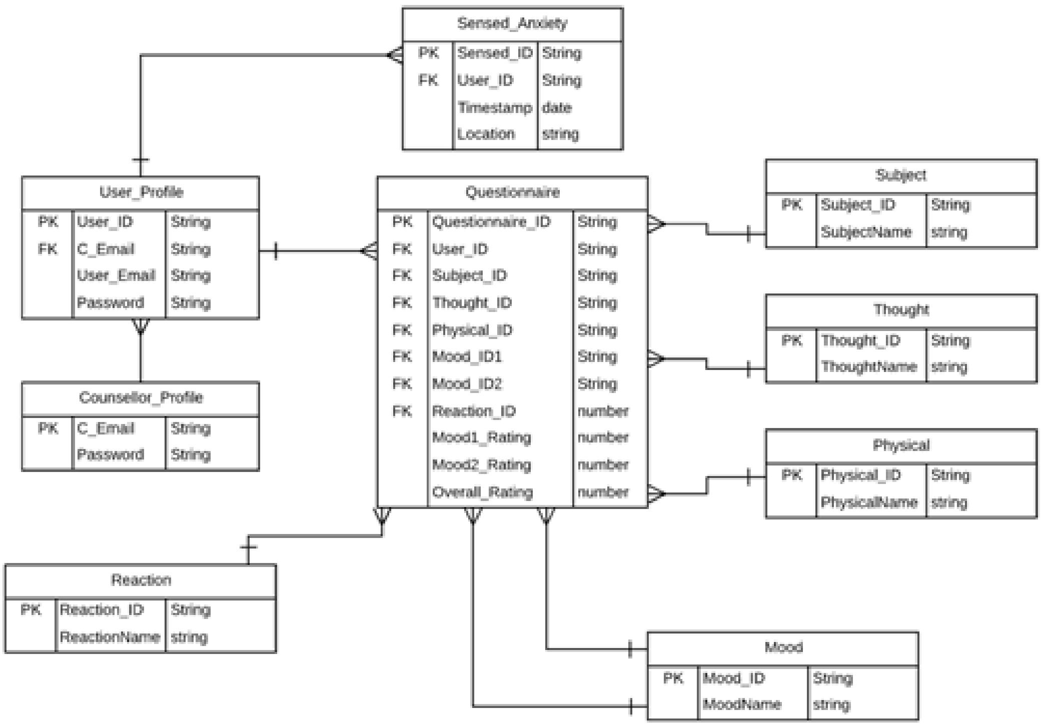


Figure 19 - Entity relational diagram for Anxiety Manager system

It was presumed that the User_Profile and Counsellor_Profile tables contain all the login information for the user and the counsellor accounts. The user table has a foreign key of the counsellor email address so the accounts can be linked. Only one counsellor can be associated with a user, but a counsellor may have many patients and therefore connected to more than one user profile.

The Questionnaire table houses all the foreign keys from the Subject, Thought, Physical, and Mood tables. These tables all contain the answers presented to the user during the questionnaire. The foreign keys in the questionnaire table indicate all the results the user has chosen. It also contains the two mood ratings and the overall mood rating result. A user may carry out many questionnaires, but only one questionnaire can be associated with one user.

The Sensed_Anxiety table incorporates the information about the anxiety episode felt by the sensors. The data stored includes the current timestamp and the location of the episode. A user can have more than one sensed anxiety episode but only one episode is associated with a user.

After some development work, it was realised that the initial design of using only remote storage would not work. Instead, during a second iteration of the database design, it was decided that a mix of remote and local storage works best for the developed system.

3.7.3 Second Iteration

The second iteration of the database design consists of two designs, one for local storage and one for remote storage.

Remote Storage

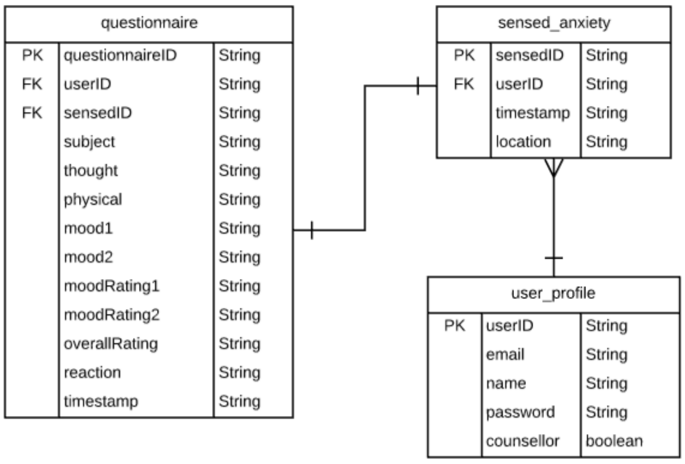


Figure 20 - Entity relational diagram for cloud storage

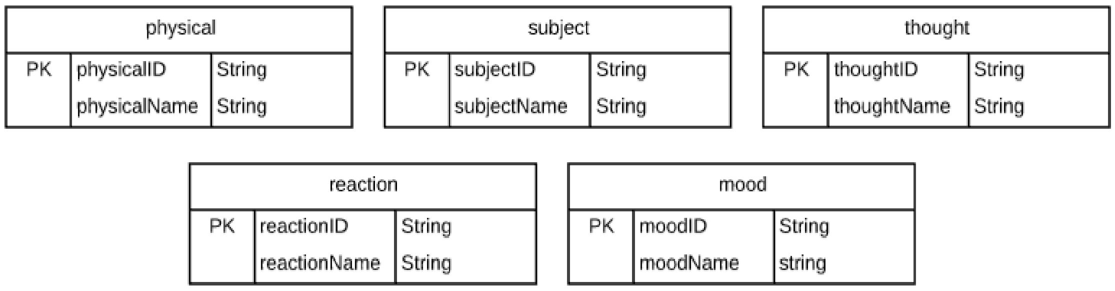
Firebase is the remote database for the developed system. After the second iteration, this database will store the data relating to questionnaires, anxiety attacks, and login (shown in Figure 20).

The questionnaire table still contains the data associated with a questionnaire, but now the data is captured directly by the user inputs in the application. Now the questionnaire table holds a sensed id and timestamp. These are either taken from a previous anxiety attack or create when a questionnaire is started. The sensed_anxiety table remains the same but now has a relationship with the questionnaire table.

The user_profile table is now a product of merging the first iteration User_Profile and Counsellor_Profile tables. Now the user_profile holds the data used for any user to log in or register. This includes the user_id, email, name, password and a boolean indicating whether or not a user is a counsellor or a regular user.

Local Storage

SQLite is the local storage for the developed system. After the second iteration, it was decided the local storage would hold the "default" questionnaire symptom responses, as well as the ones the user will add themselves. The local database design is shown below in Figure 21.



physical		
PK	physicalID	String
	physicalName	String

subject		
PK	subjectID	String
	subjectName	String

thought		
PK	thoughtID	String
	thoughtName	String

reaction		
PK	reactionID	String
	reactionName	String

mood		
PK	moodID	String
	moodName	string

Figure 21 - Diagram for local storage

The tables include a physical table, a subject table, a thought table, a mood table and a reaction table. They all have a unique id and the name of the symptom.

3.7.4 Conclusion

After some development, it was realised that the initial design for the database did not fit the architecture of the system. Following the agile methodology, a second iteration was carried out of the storage design. It was decided the storage would be split among remote and local storage. The remote storage holds the data for a completed questionnaire, anxiety attacks and login/register. The local storage holds the data used to fill in the questionnaire.

3.8 Source Code Design

3.8.1 Introduction

Before development can begin, from the user interface, technical architecture, and storage designs the source code layout needs to be created. Designing the source code will give an outlook of the system and how activities and data interact with one another. This section will discuss the class diagrams for the developed system. The source code will be divided into front-end activity classes and regular Java classes.

3.8.2 Front-End Source Code Design

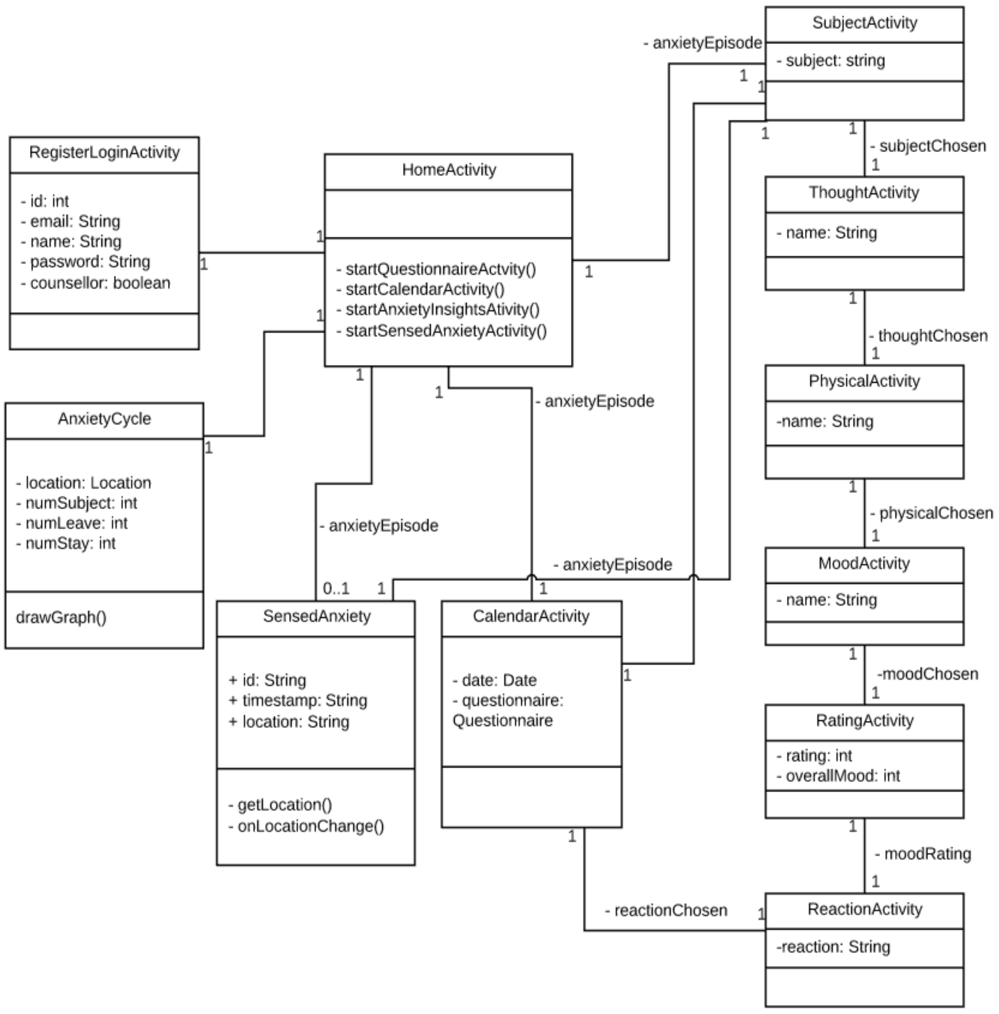


Figure 22 - Activity Class Diagram

This class diagram in Figure 22 demonstrates the activities of the developed system. All variables, return types and interactions are included. The HomeActivity interacts with all other classes. From here, SubjectActivity, CalendarActivity, SensedAnxietyActivity, AnxietyInsightActivity can be started. The HomeActivity can only be started from the Register/LoginActivity. From the SensedAnxietyActivity and the CalendarActivity, a new questionnaire can be started from the SubjectActivity. From the class diagram for the developed system, the interaction of data through the application can be modelled.

3.8.3 Java Source Code Design

The Java class model diagram can be seen in Figure 23.

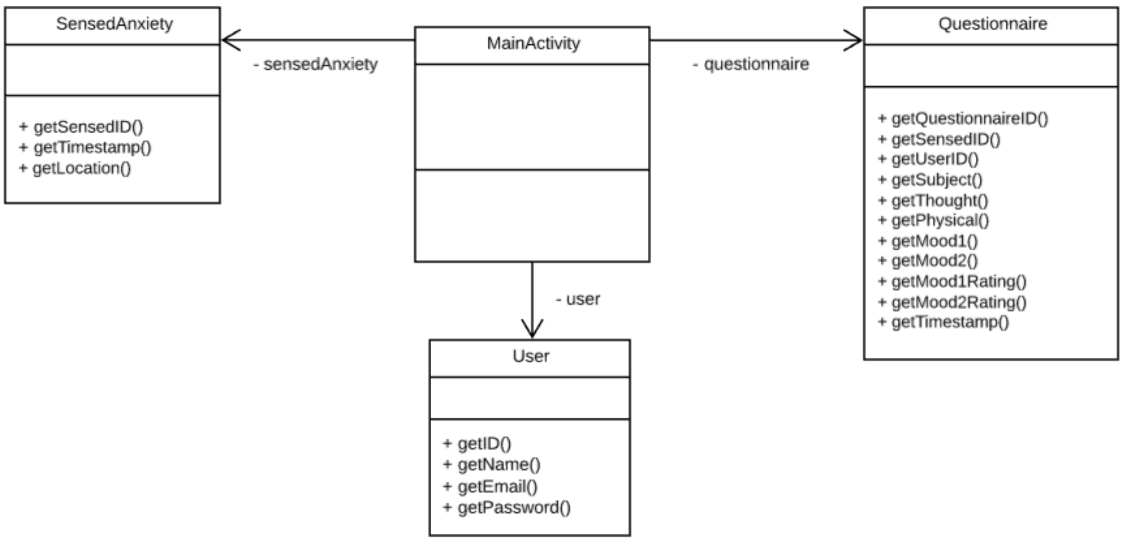


Figure 23 - Java Class model diagram

The remote storage, Firebase Database, requires all data to be accessed through objects. The Java object model diagram is like the Firebase database diagram in Figure 23. A Java object is used to access the data for questionnaire, calendar, user details and the sensed anxiety or anxiety attack list. All object classes have getters and constructors to access the data.

Chapter 4 Development

4.1 Introduction

The development of the application took place over a couple of months. Features were developed one by one during 10-14 days sprints (depending on the size of features). The order of development followed was front-end development, back-end development and middle layer development. Development work included creating user interface elements, the configuration of sensors, Arduino, and databases, as well as the construction of Java objects and Firebase queries. The source code for the application can be found here. This section will consider the development of the system.

4.2 Technical Architecture Summary

As mentioned before, the architecture of the system is a three-tier model. A summary of the architecture is shown in Figure 24. The Android application will communicate with the Arduino that is connected to the chosen sensor. The Android application will the communicate with the Firebase database through regular Java objects. The development of this system needs to meet this model.

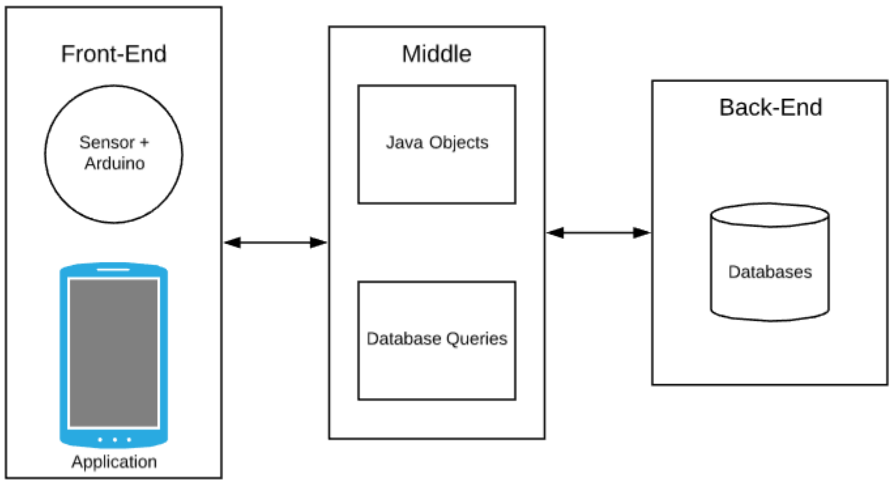


Figure 24 - Developed system architecture summary diagram

4.3 Front-End Development

4.3.1 Introduction

The presentation tier of the developed system includes interaction with the application interface and sensors. Developing the presentation layer of this application involved writing code for the user interface (based on designs) and writing code to set up the Arduino and sensor. As well as that, the Arduino and the Android needed to be assembled to communicate with one another. This section will cover the code written for the presentation layer, the challenges encountered when setting up the sensors and an eventual workaround developed for transmitting data from the sensor to the application.

4.3.2 Application Interface

This section will examine the user interface (UI) development in the Android application. The Android activities were developed using Java and XML. Some views were developed making use of external Java libraries. Each activity uses the AppCompatActivity base class and an XML layout file. Each screen interface has the same 'look and feel' by utilising the same gradient background and toolbar. The background colour of the application is blue. Blue was decided upon because it is a mentally soothing colour. "Strong blues will stimulate clear thought and lighter, soft blues will calm the mind and aid concentration." [69] The following sections look in depth at how the user interface was developed for all the features in the application.

Login and Register

For a user to gain access to the application they must register and login. The UI for the register and login page can be seen in Figure 25. The login and register layouts were developed with basic XML layouts.

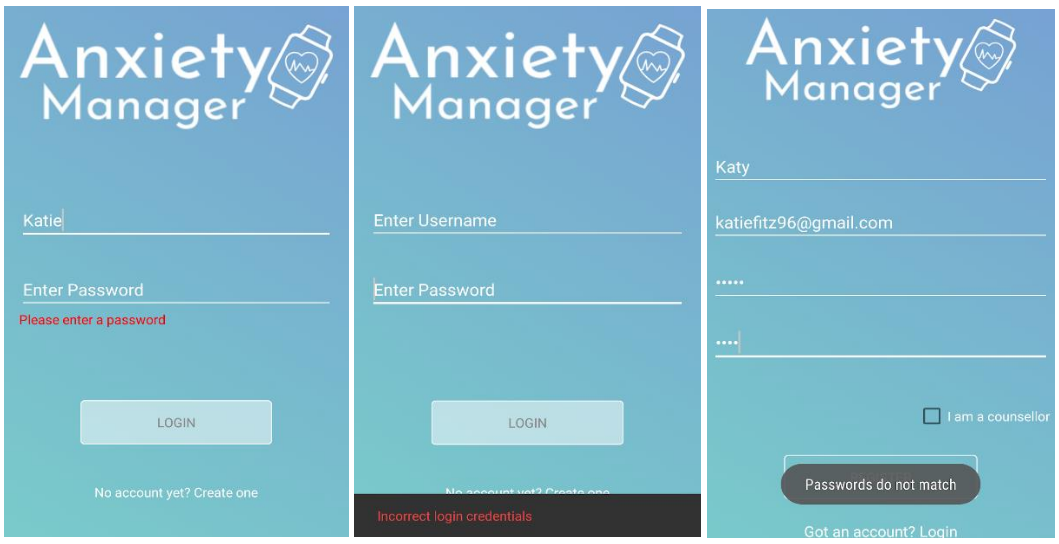


Figure 25- Login and Register interface with error prevention

The login and register activities have error prevention, Figure 25. Error prevention needs to be a core part of an application that is targeted at users of all IT skill levels. The username and password must match that in the database for a successful login. Similarly, a user's chosen password must be confirmed at the time of registration, to avoid any errors when they attempt to login later. As well as that, when a user is registering or logging in, no input can be empty. When a user is registering, they have an option to register for a counsellor profile or regular profile.

Home Screen

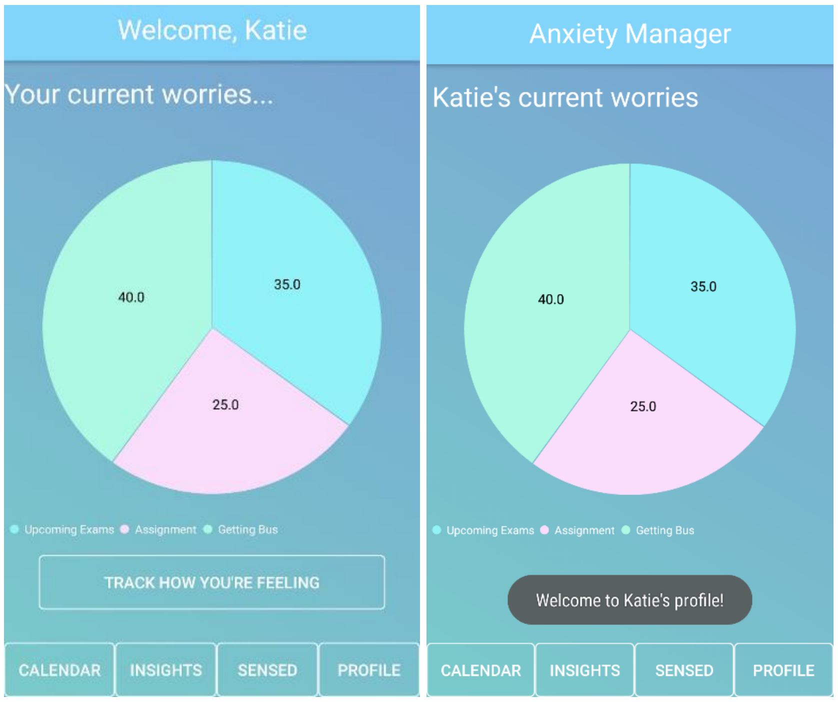


Figure 26 - Home Screen of developed system for regular users (left) and counsellor users (right)

The HomeActivity is the starting point for all other activities for users Figure 26. It is the activity that users are first navigated to following their successful login. The UI of the home activity includes a pie chart taken from the anxiety insight activity. The chart is created with the MPAndroidChart, discussed more in detail in Anxiety Insights activity section. The chart gives users a brief overview of their anxiety trends as soon as they log in. The HomeActivity for counsellors does not give an option to log anxiety questionnaires, Figure 26.

Questionnaire

The Questionnaire interface incorporates the subject, thought, physical feeling, emotion, rating and reaction activities.

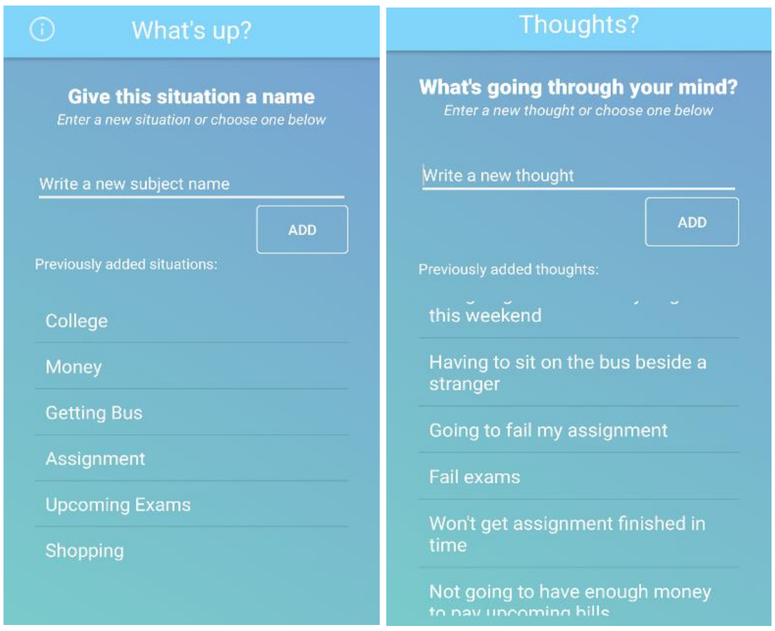

What's up?
Give this situation a name
Enter a new situation or choose one below
Write a new subject name
ADD
Previously added situations:
College
Money
Getting Bus
Assignment
Upcoming Exams
Shopping
Thoughts?
What's going through your mind?
Enter a new thought or choose one below
Write a new thought
ADD
Previously added thoughts:
this weekend
Having to sit on the bus beside a stranger
Going to fail my assignment
Fail exams
Won't get assignment finished in time
Not going to have enough money to pay upcoming bills

Figure 27 - The UI of the subject (left) and thought (right) activities

Subject and Thought Activities

The subject and thought activities share a similar UI, Figure 27. They invite users to input a subject and thought about the current anxiety attack. Users can input a new response using the input box or can choose from a default or previous response displayed in a list view below the input box. If a user chooses to input a new response, it is added to the list and they are automatically navigated to the next question. If a user chooses a default or previously added response, they are automatically taken to the next question.

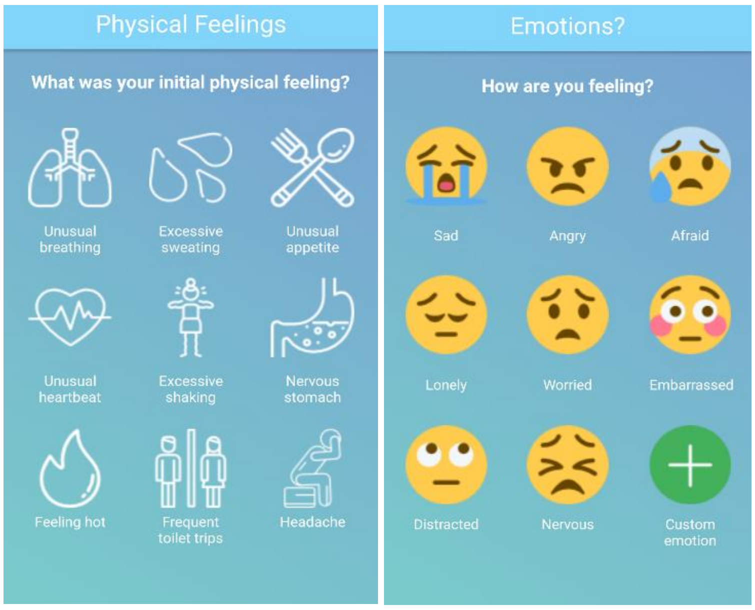


Figure 28 - The UI of the physical feelings (left) and emotion (right) activities

Physical Feeling and Emotion Activities

The questionnaire physical feeling and emotion activities also share a similar UI. They consist of clickable icons for users to log their response, displayed in Figure 28. The clickable icons are from a canonical perspective to allow users to quickly recognize them [64]. Users can only log one physical symptom per anxiety episode, their initial physical feeling. Following user feedback, it was established that users did not want to be limited to choosing only the represented emotions in the application and expressed a need for a way to input different feelings. A custom emotion view was developed as a result of this feedback. When a user wishes to add a custom emotion, a dialog opens and users can input a new emotion or choose from a previously added custom emotion, shown in Figure 29. Users can choose two emotions or one custom emotion per questionnaire. The code snippet below shows how the custom emotion dialog was set up.

```
AlertDialog.Builder otherDialog = new AlertDialog.Builder(EmotionActivity.this);
final View dialogView = getLayoutInflater().inflate(R.layout.mood_dialog, null);
final EditText addOther = dialogView.findViewById(R.id.otherEmotion);
final ListView emotionList = dialogView.findViewById(R.id.emotionList);
Button add = dialogView.findViewById(R.id.add);
```

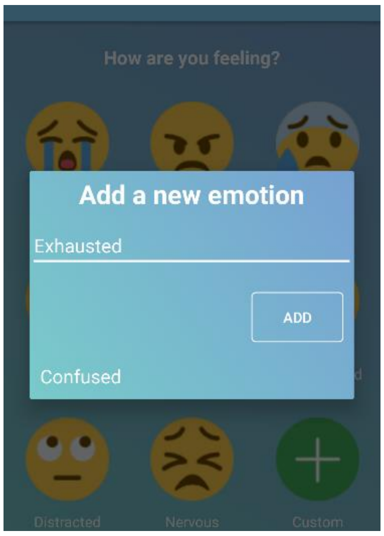


Figure 29 - Custom emotion dialog

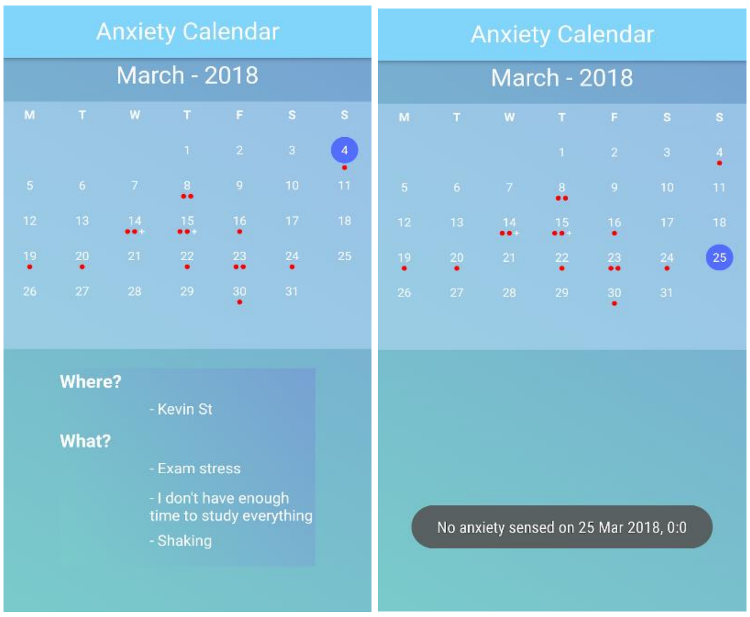
Rating Mood Activity

When a user has chosen the associated emotions of their anxiety attack, they are then asked to submit the severity of those emotions. They are navigated to the rating. The original design included identifying the previously chosen emotions in the rating activity, by their icons. Challenges were encountered with passing the chosen emotions to the rating activity and then displaying the related icon. As well as that, the developed application would not be capable of generating an icon for every custom emotion inputted by users. A decision was made to display the emotion text instead of the icons to incorporate any emotion chosen or added. The emotions are rated using sliders. Users have the choice of choosing one custom emotion or two default emotions. As well as rating emotions, users are asked to rate their overall mood during the anxiety attack. This value is captured to use for the user's anxiety trends.

Reaction Activity

Following the rating activity, users are navigated to the final questionnaire activity to log their reaction to the anxiety attack. Much like the physical feeling and emotion activities, this activity uses only clickable icons for users to communicate their response. The reaction activity asks users to track whether they stayed in the situation or left. Once a user has decided on their response, they must click the 'done' button to end the questionnaire and submit it to the application. This is the final activity of the questionnaire.

Calendar



Where?

- Kevin St

What?

- Exam stress
- I don't have enough time to study everything
- Shaking

No anxiety sensed on 25 Mar 2018, 0:0

Figure 30 - Calendar Activity User Interface

Users can view their sensed anxiety attacks in a calendar view shown in Figure 30. As well as that, counsellors have access to their patients calendar, but they do not have permission to log questionnaires from calendar. In this activity, users can view their previous anxiety episodes in a calendar layout. The calendar UI is developed using the CompactCalendarView. The view is a "simple calendar view which provides scrolling between months" [55]. The decision to use this view was based on the API queries and listeners the calendar provides for specific dates and month changes [55]. As well as that, the CompactCalendarView enables an icon to be displayed underneath a date when an event occurs on that date. The following code is used to add an icon to the calendar for an event.

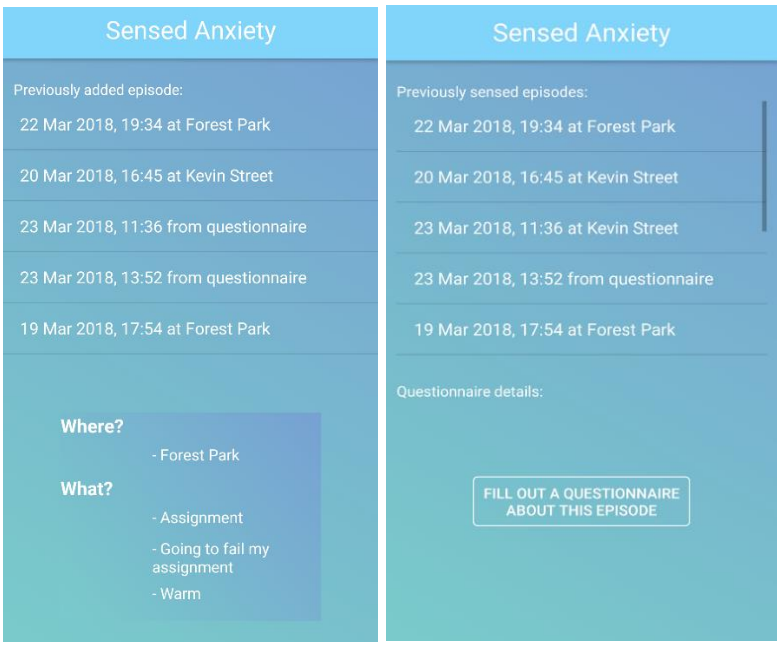
```
compactCalendar.shouldDrawIndicatorsBelowSelectedDays(true);
```

The date of the anxiety attack determines where the event is created in the calendar view. The timestamp from the users sensed anxiety is used to populate the calendar with events shown in the code below.

```
    if (sensedAnxiety.getTimestamp() != null) {

        //get timestamps from sensed anxiety object
        timestamps.add(Long.parseLong(sensedAnxiety.getTimestamp()));

    }

}

if (!timestamps.isEmpty()){

    for(int i = 0; i < timestamps.size(); i++){

        compactCalendar.addEvent(new Event(Color.RED, timestamps.get(i), "Anxiety Episode"));

    }
}
```

An onClick listener can then be added, to determine which date has been clicked. When a user clicks on a date, the questionnaire for it is loaded from the database underneath the calendar, or the user is invited to log a questionnaire via the questionnaire button.

Sensed Anxiety



Sensed Anxiety

Previously added episode:

22 Mar 2018, 19:34 at Forest Park

20 Mar 2018, 16:45 at Kevin Street

23 Mar 2018, 11:36 from questionnaire

23 Mar 2018, 13:52 from questionnaire

19 Mar 2018, 17:54 at Forest Park

Where?
- Forest Park

What?
- Assignment
- Going to fail my assignment
- Warm

Sensed Anxiety

Previously sensed episodes:

22 Mar 2018, 19:34 at Forest Park

20 Mar 2018, 16:45 at Kevin Street

23 Mar 2018, 11:36 at Kevin Street

23 Mar 2018, 13:52 from questionnaire

19 Mar 2018, 17:54 at Forest Park

Questionnaire details:

Figure 31- Sensed Anxiety Activity User Interface

The sensed anxiety activity interface, Figure 31, consists of a simple list view of previous anxiety attacks from questionnaires and sensed anxiety. As well as that two invisible UI elements, a button and a questionnaire details layout are present. The list view is populated with sensed anxiety elements from the database for a specific user. If a questionnaire was logged for an anxiety episode, the questionnaire details layout would be made visible (with the code below) and shown when a user clicks on the corresponding list item.

```
questionnaireStart.setVisibility(View.INVISIBLE);
questionnaireDetails.setVisibility(View.VISIBLE);
```

If there is no questionnaire for that sensed episode, the fill out questionnaire button is made visible (with the code below) and the user is invited to log one for it. Pressing the "Fill out a questionnaire" button view in Figure 34, starts a new questionnaire for the sensed episode.

```
questionnaireDetails.setVisibility(View.INVISIBLE);
questionnaireStart.setVisibility(View.VISIBLE);
```

Counsellors have access to their patients sensed anxiety list, but they do not have permission to log questionnaires from the list

Anxiety Insights

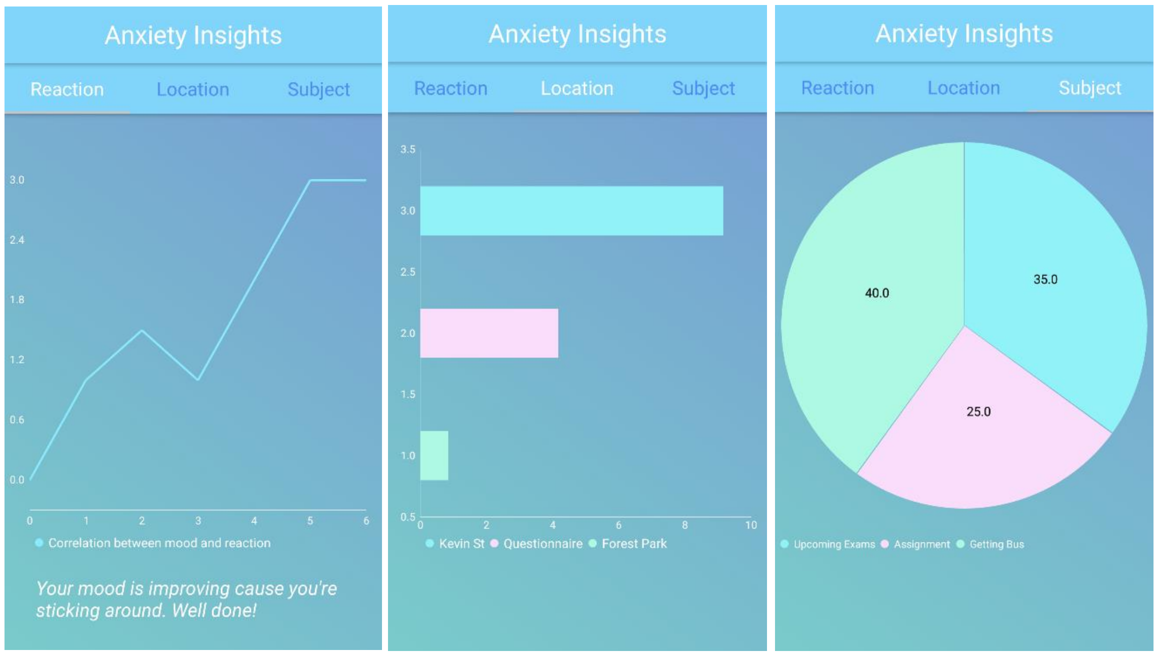
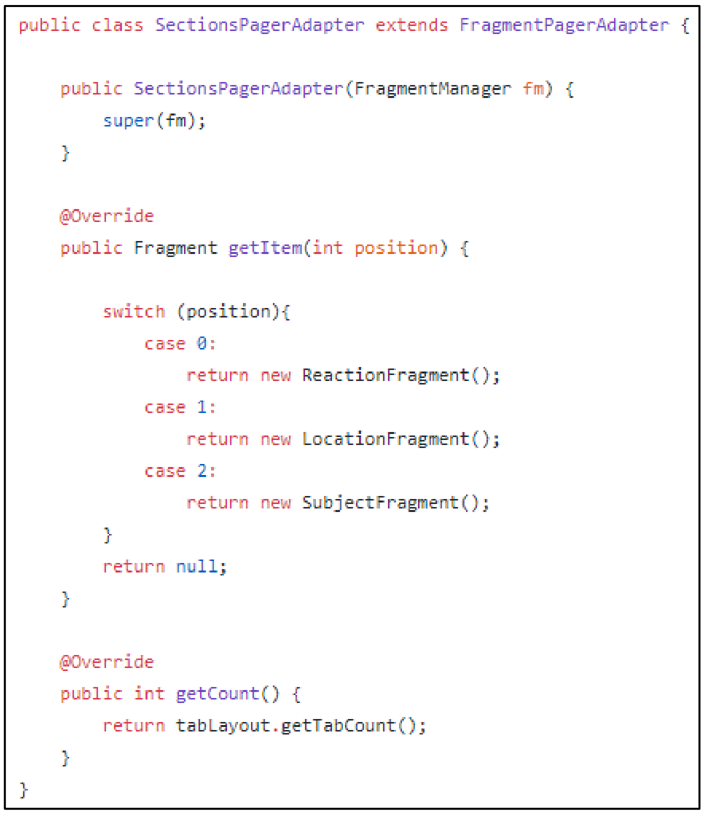


Figure 32 - Anxiety Insights Activity User Interface

The anxiety insight activity UI was developed using Android Fragments, TabLayout and ViewPager to provide slide transitions within the activity shown in Figure 32. The slide transition enables users to effortlessly navigate through their anxiety trends. Counsellors have access to their patients anxiety insight. The code below shows the SectionsPagerAdapter which allows the application to provide the correct fragment per tab. Depending on which tab is chosen, the application will return the appropriate fragment object.



```
public class SectionsPagerAdapter extends FragmentPagerAdapter {

    public SectionsPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {

        switch (position){
            case 0:
                return new ReactionFragment();
            case 1:
                return new LocationFragment();
            case 2:
                return new SubjectFragment();
        }
        return null;
    }

    @Override
    public int getCount() {
        return tabLayout.getTabCount();
    }
}
```

The trends are provided to the user through three different charts as previously mentioned. The charts were developed using the MPAndroidChart library [56]. It is possible to draw custom charts in Java using the OnDraw method and Canvas object, but the MPAndroidChart provides more structure and consistency to the developed system's charts and gives them all the same look and feel. The library also offers legends that correspond to the chart, which would not look seamless developed without the help of the library [56]. The code below shows how the bar chart and legend was drawn for the location fragment.

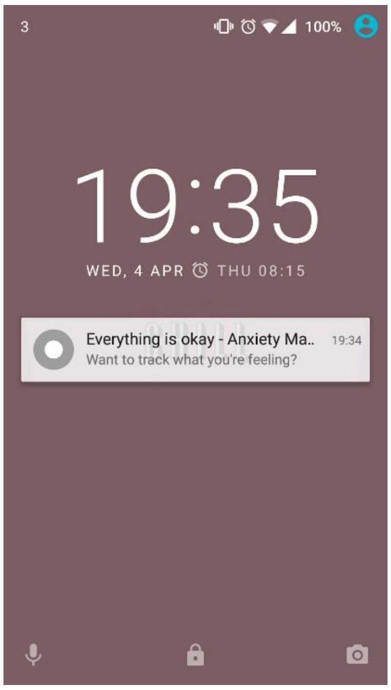
```
BarDataSet barDataSet = new BarDataSet(points, "");
barDataSet.setDrawValues(false);

//add color to dataset
ArrayList<Integer> color = new ArrayList<>();
color.add(Color.rgb(145, 243, 247));
color.add(Color.rgb(250, 220, 251));
color.add(Color.rgb(174, 249, 228));

barDataSet.setColors(color);

List<LegendEntry> entries = new ArrayList<>();

for (int i = 0; i < labels.length; i++) {
    LegendEntry entry = new LegendEntry();
    entry.formColor = color.get(i);
    entry.label = labels[i];
    entries.add(entry);
}

Legend legend = locationChart.getLegend();
legend.setForm(Legend.LegendForm.CIRCLE);
legend.setTextColor(Color.rgb(255, 255, 255));
legend.setTextSize(12);
legend.setCustom(entries);

BarData barData = new BarData(barDataSet);
barData.setBarWidth(0.4f);
```

The Java code below sets up the data and style for the bar chart from the dataset previously produced.

```
locationChart.getDescription().setEnabled(false);
locationChart.setDrawMarkers(false);
locationChart.setScaleEnabled(false);
locationChart.setPinchZoom(false);
locationChart.setDoubleTapToZoomEnabled(false);
locationChart.setData(barData);
```

Notification

The notification feature of the developed system alerts users during anxiety attacks and once a day. Once the sensor value goes over the set threshold, the application will send a notification (shown in Figure 33) to the user to invite them to track a questionnaire about. Users are also reminded once a day to track any anxiety that happened that day. By tracking consistently users will reap the benefits of the system.



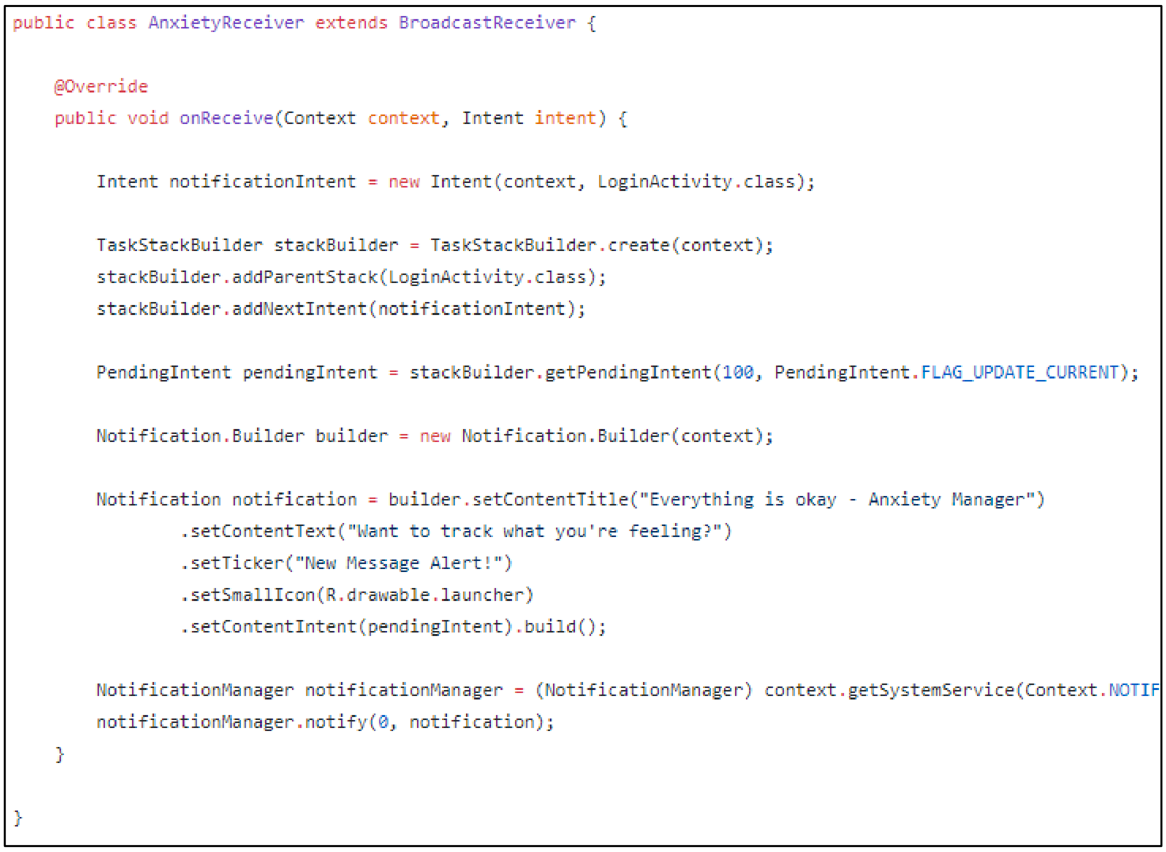
Everything is okay - Anxiety Ma..
Want to track what you're feeling?

Figure 33 - Sensed Anxiety notification

The development of notifications for Android devices involves using a BroadcastReceiver class to build notifications and passing them to an AlarmManager class. The AlarmManager will receive the time of when to push the notification. From there the notification is built in the BroadcastReceiver.onReceive() method and passed back and triggered. The AlarmManager class lets the application to run some scheduled event. Alarms are "retained while a device is asleep" [57] this makes them perfect for scheduling notifications. The snippet below shows how the AlarmManager object is created and used with the broadcastReceiver to send the notification.

```java
private void sendNotification() {

    AlarmManager alarmManager = (AlarmManager) getSystemService(Context.ALARM_SERVICE);

    Intent notificationIntent = new Intent(this, AnxietyReceiver.class);

    PendingIntent broadcast = PendingIntent.getBroadcast(this, 100, notificationIntent, PendingInt

    Calendar cal = Calendar.getInstance();

    if (alarmManager != null) {
        alarmManager.setExact(AlarmManager.RTC_WAKEUP, cal.getTimeInMillis(), broadcast);
    }
}
```

The BroadcastReceiver class is used when creating the notification intent. The onReceive method builds the notification and sets the activity started when the user opens the application from the notification. The code belows indicates how the notification is built.



```
public class AnxietyReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {

        Intent notificationIntent = new Intent(context, LoginActivity.class);

        TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);
        stackBuilder.addParentStack(LoginActivity.class);
        stackBuilder.addNextIntent(notificationIntent);

        PendingIntent pendingIntent = stackBuilder.getPendingIntent(100, PendingIntent.FLAG_UPDATE_CURRENT);

        Notification.Builder builder = new Notification.Builder(context);

        Notification notification = builder.setContentTitle("Everything is okay - Anxiety Manager")
                .setContentText("Want to track what you're feeling?")
                .setTicker("New Message Alert!")
                .setSmallIcon(R.drawable.launcher)
                .setContentIntent(pendingIntent).build();

        NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.NOTIF
        notificationManager.notify(0, notification);
    }

}
```

Choose A Patient

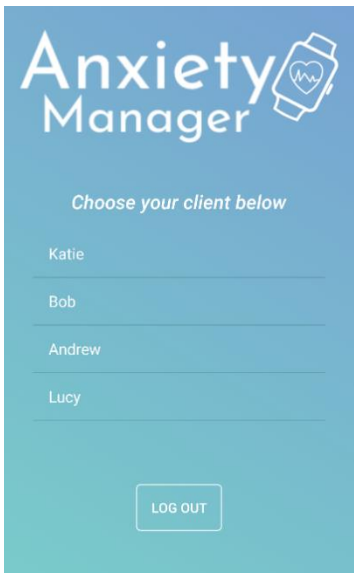


Figure 34 - Choose a Patient list view

This feature was the last on the priority list. It is not essential for counsellors to have a profile, and it makes no direct effect on a regular user's personal anxiety understanding whether their counsellor has a view of their profile or not. As a result of time running out, the counsellor profile couldn't be developed as fully intended. The counsellor profile in the developed system currently displays all registered users. The 'Choose A Patient' activity (Figure 34) is the first screen after the counsellor logs in. Here counsellors are presented with a list of registered users and once they choose a name they will be able to view the profile of that individual.

4.3.3 Arduino and Sensors

This section will look at the Arduino Uno board and the pulse and Grove GSR sensors configuration. The proposed method for monitoring changes in physical symptoms with the sensor is that a threshold will be assigned for unusual physical symptoms. Once the incoming value is above that threshold the user will be alerted with a notification.

The Arduino and sensors were set up using the Arduino IDE and code tutorials from the Grove GSR Sensor wiki [58] and, Pulse Sensor code and guide tutorial [59]. Because of issues with the Arduino to Android connection, a workaround had to be developed to mock the incoming sensor data to the developed application.

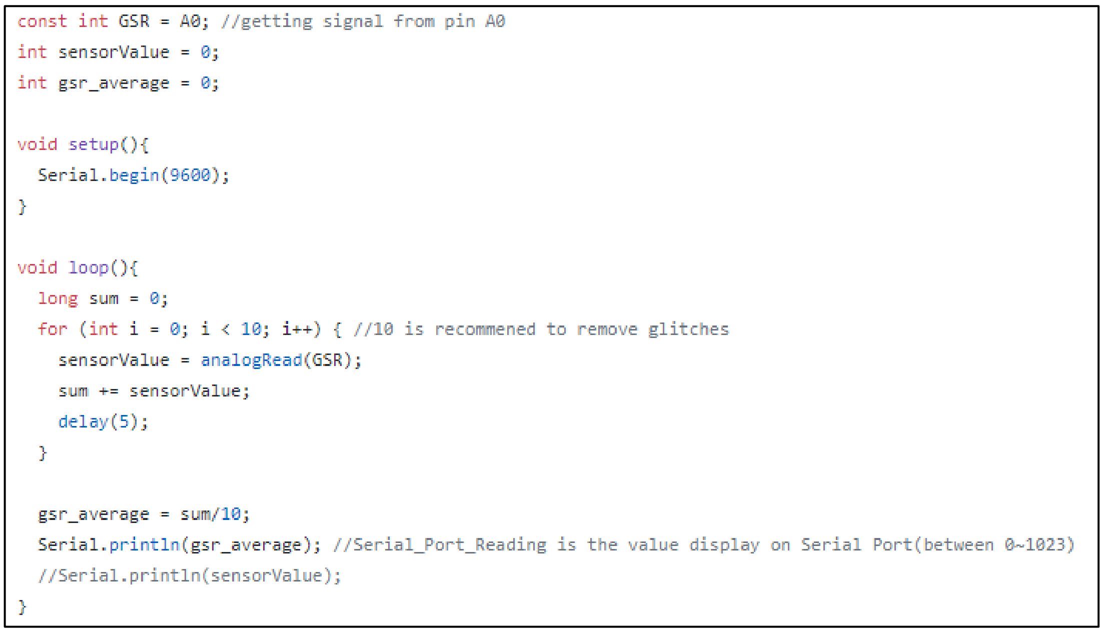
Arduino Configuration

As mentioned previously, two sensors were obtained for development to combat any potential issues. This meant two versions of code were written, one for the pulse sensor and one for the Grove GSR sensor.

Firstly, the following snippet was uploaded to the Arduino Uno for configuring the pulse sensor. It is the code from the pulsesensor.com “Code and Guide” tutorial for setting up an Arduino Uno to read data from a pulse sensor [59]. The readings can be printed out to the serial plotter for testing.

```
int PulseSensorPurplePin = 0;     // Pulse Sensor PURPLE WIRE connected to ANALOG PIN 0

int Signal;                // holds the incoming raw data. Signal value can range from 0-1024

// The SetUp Function:
void setup() {
    Serial.begin(9600);         // Set's up Serial Communication at certain speed.

}

// The Main Loop Function
void loop() {

    Signal = analogRead(PulseSensorPurplePin);  // Read the PulseSensor's value.
                                                // Assign this value to the "Signal" variable.

    Serial.println(Signal);                    // Send the Signal value to Serial Plotter.

    delay(10);


}
```

After testing the code for the pulse sensor, the following code was uploaded to the Arduino Uno for the Grove Sensor configuration. It is the code from the wiki.seeedstudio.com Grove-GSR Sensor tutorial for setting up an Arduino to read data from the GSR sensor. The signal is read from pin A0, and the average reading is calculated every ten measurements. This removes any inaccurate readings. The readings can then be printed out to the serial plotter for testing.



```
const int GSR = A0; //getting signal from pin A0
int sensorValue = 0;
int gsr_average = 0;

void setup(){
  Serial.begin(9600);
}

void loop(){
  long sum = 0;
  for (int i = 0; i < 10; i++) { //10 is recommened to remove glitches
    sensorValue = analogRead(GSR);
    sum += sensorValue;
    delay(5);
  }

  gsr_average = sum/10;
  Serial.println(gsr_average); //Serial_Port_Reading is the value display on Serial Port(between 0~1023)
  //Serial.println(sensorValue);
}
```

Once the code was uploaded to the Arduino board, the sensors could be installed and tested.

Sensor Configuration

The first sensor to be set up was the pulse rate sensor. Firstly, the pulse sensor code was uploaded to the Arduino. Additionally, the blue wire on the pulse sensor was connected to A0 pin on the Arduino Uno board. As well as, the purple wire to the GND pin and the green wire to the 5v pin. The pulse sensor was connected to the user's finger and tested as shown below in Figure 35.

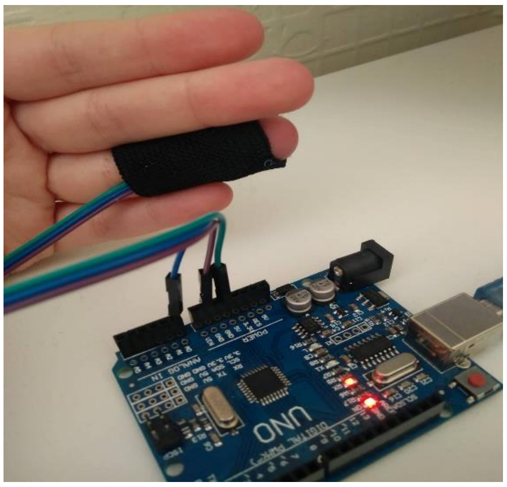


Figure 35 - The pulse sensor connected to user's finger

Unfortunately, no meaningful values were obtained from the pulse sensor as seen in Figure 36. This pulse sensor was quite delicate, and it's plausible that too much pressure was placed on the sensor when it was being connected to the elastic band. As this issue was foreseen, development continued with the Grove sensor only.

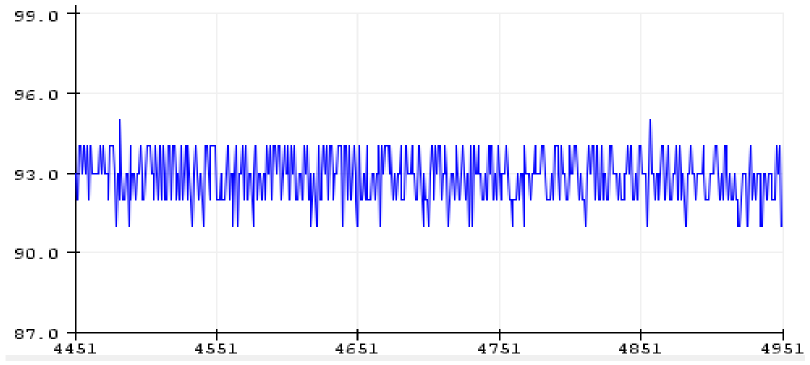


Figure 36 - Values from pulse sensor

The Grove sensor code was uploaded to the Arduino, these steps were followed to configure the sensor:

1. The Grove sensor was connected to the Arduino as shown in Figure 39 below.
    a. Black wire to GND pin
    b. Red wire to 5v pin
    c. Yellow wire to A0 pin
2. The two finger electrodes were connected to the Grove sensor as shown in Figure 37.
3. The Arduino was then connected to a computer using the USB connection.
4. Then, using the serial plotter in the Arduino IDE, the resistance trends were tested.

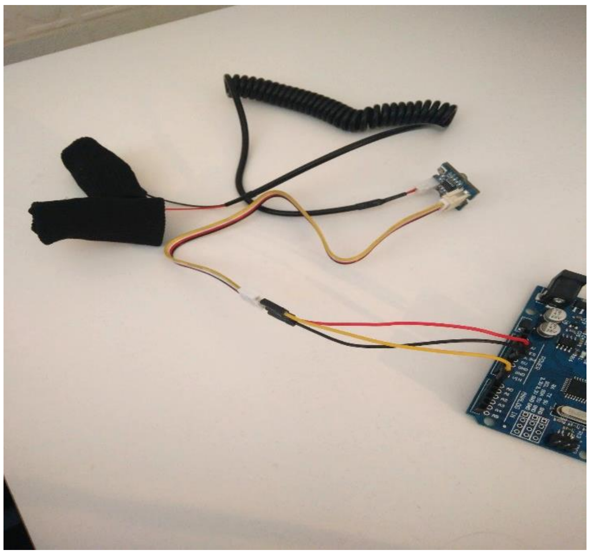


Figure 37 - Connecting GSR to Arduino

The Grove sensor measures electric resistance or how the user's skin resists the flow of electric current. Therefore, any changes in physical symptoms or emotion are displayed by a decrease in values. Shown in Figure 38 is the serial plotter when no resistance is being measured. In Figure 39 is the serial plotter diagram when resting. It is vital to note, that this resting value is inaccurate. It is hard for users to not become conscious of the sensor monitoring them and whether they are resting "accurately" or not. This initial test indicated that this could be an issue for the application to detect authentic anxiety if users are aware of the sensors.

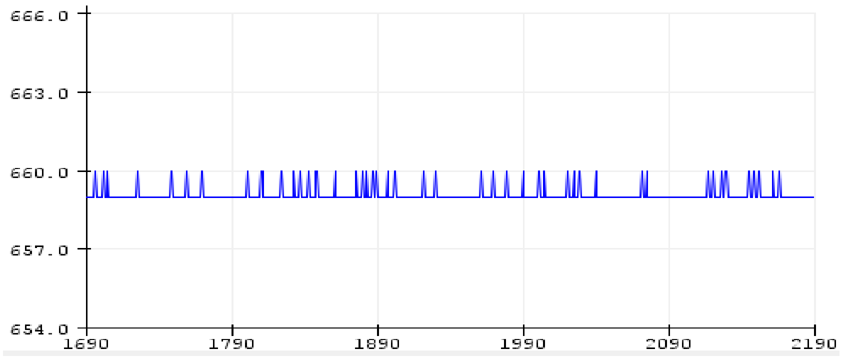


Figure 38 - when no user is attached to sensor

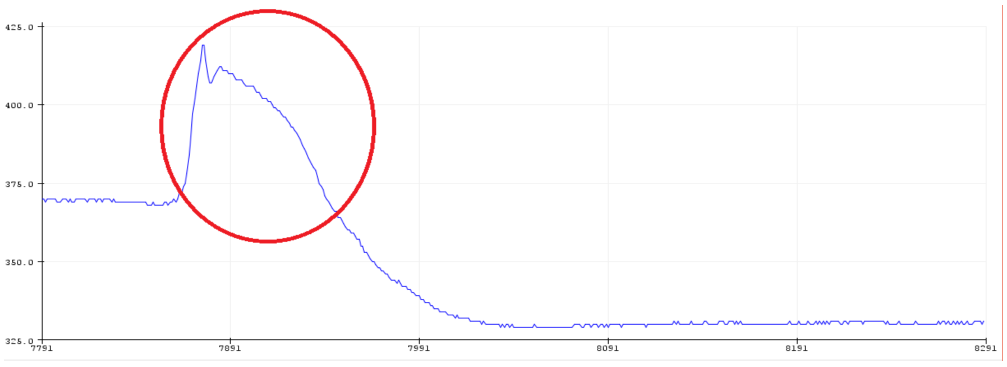


Figure 39 - When user is resting but becomes aware of sensor

Arduino Android Connection

Following the Arduino and Grove sensor set up, the Arduino needed to be connected to the Android device. It was intended the values from the Arduino would be sent to the application from the USB connection, shown in Figure 40. This involves the use of a USB On-The-Go (OTG) adapter. The USB OTG allows the Android phone to host other devices like an Android.

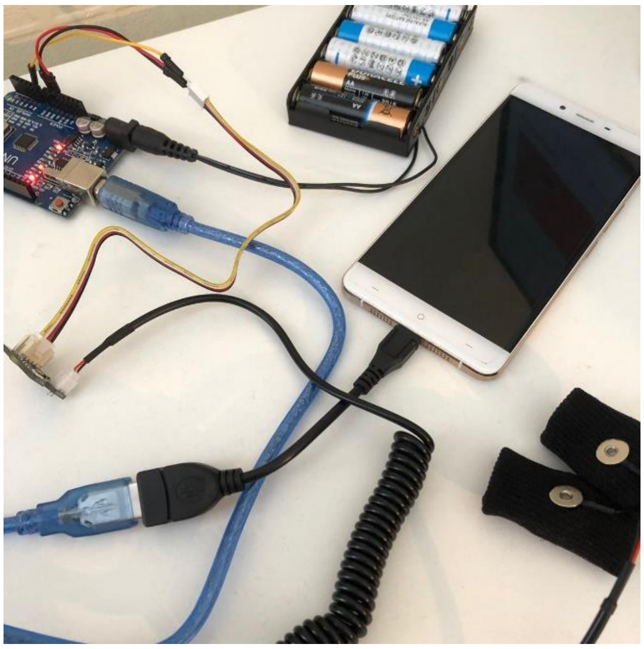


Figure 40 - Full Arduino to Android device connection

It was expected that the Android application would implement the class and an external library called OmarAflak/Arduino-Library. This is a library that allows an Android application “communicates with Arduino through USB” [60]. The following code snippet was used to connect the Arduino and Android application. The code snippet was coded following “Cause You're Stuck: Connect Arduino to Android through USB tutorial” [61].

```
<uses-feature android:name="android.hardware.usb.host" />
```

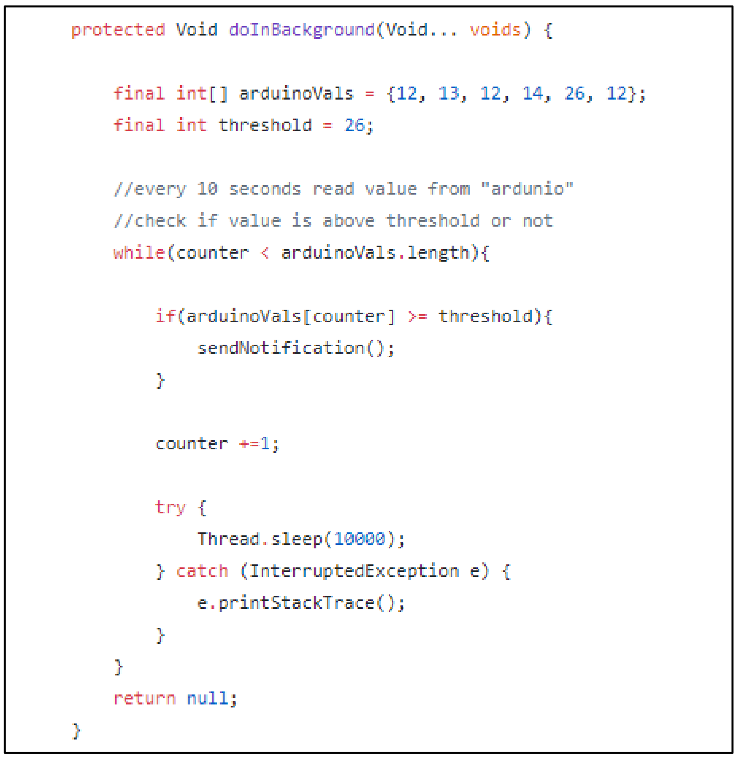
```
Intent intent = getIntent();
String action = intent.getAction();

UsbDevice device = intent.getParcelableExtra(UsbManager.EXTRA_DEVICE);
if (UsbManager.ACTION_USB_DEVICE_ATTACHED.equals(action)) {

    Arduino arduino = new Arduino(this);

    arduino.setArduinoListener(new ArduinoListener(){

        @Override
        protected void onStart() {
            super.onStart();
            arduino.setArduinoListener(this);
        }

        @Override
        protected void onDestroy() {
            super.onDestroy();
            arduino.unsetArduinoListener();
            arduino.close();
        }

        @Override
        public void onArduinoAttached(UsbDevice device) {
            display("Arduino attached!");
            arduino.open(device);
        }
```

There were some challenges with connecting the Arduino and the Android application. The device did not support data transfer across the USB OTG adapter and the Arduino was not registering with the Android device. As well as that, the library OmarAflak/Arduino-Library does not support receiving data from an Arduino. To save time and to prove the project concept is valid, a workaround was decided on to overcome the issues.

Sensor Connection Workaround

To overcome the issues with the Arduino and Android connection, the data coming from the Arduino needed to be mocked.

To simulate sensor data incoming over time, an AsyncTask (a helper class that performs background operations) method runs in the background on the HomeActivity. Firstly, an array is initialized with some values and a threshold is decided on. The array simulates the values that would come from the Arduino. The doInBackground method allows the application to read the values from the array, much like it would read the values from the Arduino. Using a thread.sleep method, the reading of the array can be paused by 10 seconds. This means the values of the array are read at an interval of once every 10 seconds. If the value read from the Arduino array is greater than or equal to the threshold, then this indicates that there is a change in physical symptoms. From that, a notification can be sent to the user and the location can be captured. The code snippet below shows how the application checks the Arduino array.



```
protected Void doInBackground(Void... voids) {

    final int[] arduinoVals = {12, 13, 12, 14, 26, 12};
    final int threshold = 26;

    //every 10 seconds read value from "ardunio"
    //check if value is above threshold or not
    while(counter < arduinoVals.length){

        if(arduinoVals[counter] >= threshold){
            sendNotification();
        }

        counter +=1;

        try {
            Thread.sleep(10000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
    return null;
}
```

4.3.4 Conclusion

The initial idea that the Android device would communicate with the Arduino and sensor was changed when issues arose with the physical connection between the Android mobile and Arduino. Once the UI and the sensor workaround were established for simulating the back-end of the system needed to be created to store the data from the front-end.

4.4 Back-End Development

4.4.1 Introduction

As previously mentioned, the back-end of the developed system will consolidate a combination of remote and local storage. Before the Android application can interact with the remote and local databases, the databases need to be set up. This section will cover how the Firebase and SQLite databases were configured.

4.4.2 Remote Database

Firebase Set Up

Before any data could be pushed to the cloud, the Firebase database had to be configured for the developed application. Firstly, a project was created in the Firebase console. Following that, the developed Android application needed to be added to the newly created project, Figure 41.

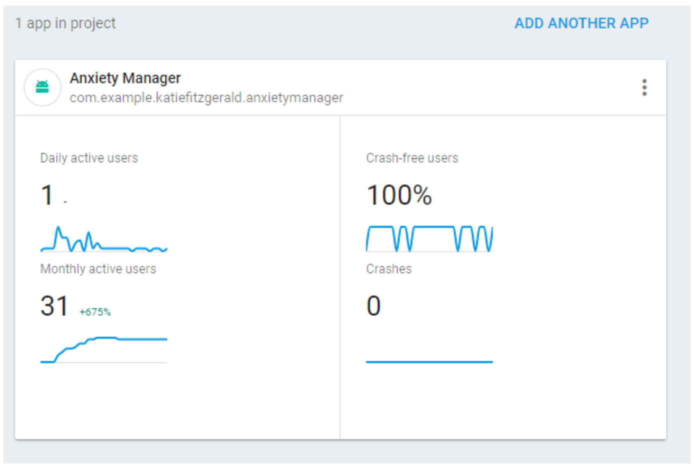


Figure 41 - The application added to the Firebase console

The application package name and SHA-1 certificate were inserted to the console and a google-services.json file was generated and added to the application. After that, the following dependencies were added to the application build.gradle file.

```
compile 'com.google.firebase:firebase-database:11.8.0'
```

The application was now ready to add database references.

Firebase References

Before data could be added to the database, references to the database and its child nodes needed to be created. As Firebase is a NoSQL database, it uses a JSON format to store information. Data is stored as JSON object nodes, with subsequent child nodes. When you reference the Firebase database, you are getting an instance of the desired node. The below code is the import statements used to reference the Firebase database dependency.

```java
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
```

The following code shows how the sensed anxiety node was referenced in the Firebase Anxiety Manager database. If the node does not exist, then one is created when the application tries to commit data to it.

```java
DatabaseReference SensedAnxietyDB = FirebaseDatabase.getInstance().getReference("sensed_anxiety");
```

4.4.3 Local Database

SQLite Set up

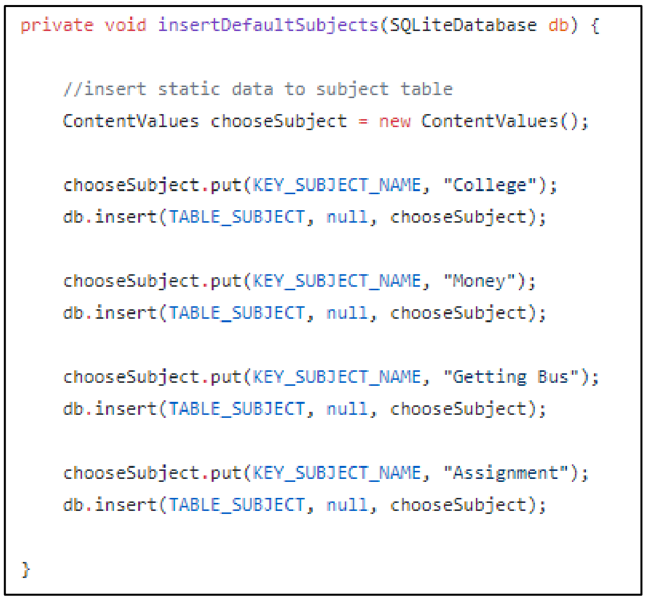
The SQLite database was created with a database helper class that extends the SQLiteOpenHelper. The SQLiteOpenHelper "helps in opening SQLiteDatabase objects and in manging database upgrades" [62]. This helper is a straightforward way to open, insert into, delete and close databases on the local Android device. The code snippet below is the constructor for my database helper class.

```java
private static class MyDatabaseHelper extends SQLiteOpenHelper {

    public MyDatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
```

The following code is the onCreate method, that is executed when the application is first installed on the Android device. These function calls, executes the create statements for the tables.

```java
db.execSQL(CREATE_SUBJECT_TABLE);
db.execSQL(CREATE_THOUGHT_TABLE);
db.execSQL(CREATE_PHYSICAL_TABLE);
db.execSQL(CREATE_MOOD_TABLE);
db.execSQL(CREATE_REACT_TABLE);
db.execSQL(CREATE_USER_TABLE);
```

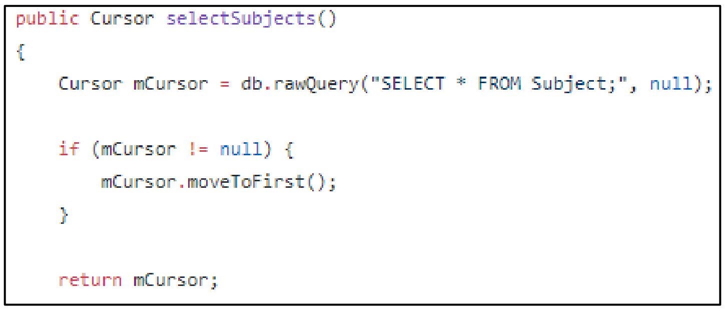
The default questionnaire responses get inserted into the local database when a user first installs the application. Below is the standard code to insert default values into a table.



```
private void insertDefaultSubjects(SQLiteDatabase db) {

    //insert static data to subject table
    ContentValues chooseSubject = new ContentValues();

    chooseSubject.put(KEY_SUBJECT_NAME, "College");
    db.insert(TABLE_SUBJECT, null, chooseSubject);

    chooseSubject.put(KEY_SUBJECT_NAME, "Money");
    db.insert(TABLE_SUBJECT, null, chooseSubject);

    chooseSubject.put(KEY_SUBJECT_NAME, "Getting Bus");
    db.insert(TABLE_SUBJECT, null, chooseSubject);

    chooseSubject.put(KEY_SUBJECT_NAME, "Assignment");
    db.insert(TABLE_SUBJECT, null, chooseSubject);

}
```

SQLite References

The following section outlines how data selected from the SQLite local database and used to populate the front-end elements. The data is mostly used in lists across the application. The code snippet below shows how the default values are then selected from a table. The select query returns a Cursor object as displayed below.



```
public Cursor selectSubjects()
{
    Cursor mCursor = db.rawQuery("SELECT * FROM Subject;", null);

    if (mCursor != null) {
        mCursor.moveToFirst();
    }

    return mCursor;
```

For the cursor to be displayed in the subject list a list cursor adapter needs to be used. A list adapter is a “ bridge between a ListView and the data that backs the list” [63]. Sometimes custom adapters need to be created for certain views that are not simple, or for data that is not simple text, such as a cursor. The following code snippet is from a custom cursor adapter that is used to populate the default subject list.

```
public View newView(Context context, Cursor cursor, ViewGroup parent) {
    return LayoutInflater.from(context).inflate(R.layout.list_view_questionnaire, parent, false);
}

//bind data to the given view
@Override
public void bindView(View view, Context context, Cursor cursor) {


    TextView subjectName = (TextView) view.findViewById(R.id.textView);

    // Extract properties from cursor
    String subject_name = cursor.getString(cursor.getColumnIndexOrThrow("SubjectName"));

    // Populate fields with extracted properties
    subjectName.setText(subject_name);
}
```

4.4.4 Conclusion

The combination of the remote and local storage meant the data could be separated. The local database stores the default values in the application and the remote database will hold the anxiety related data. Once the databases were configured, data could be added. The SQLite data was added during the set-up, so the next section will explore the remote databases queries.

4.5 Middle Layer Development

4.5.1 Introduction

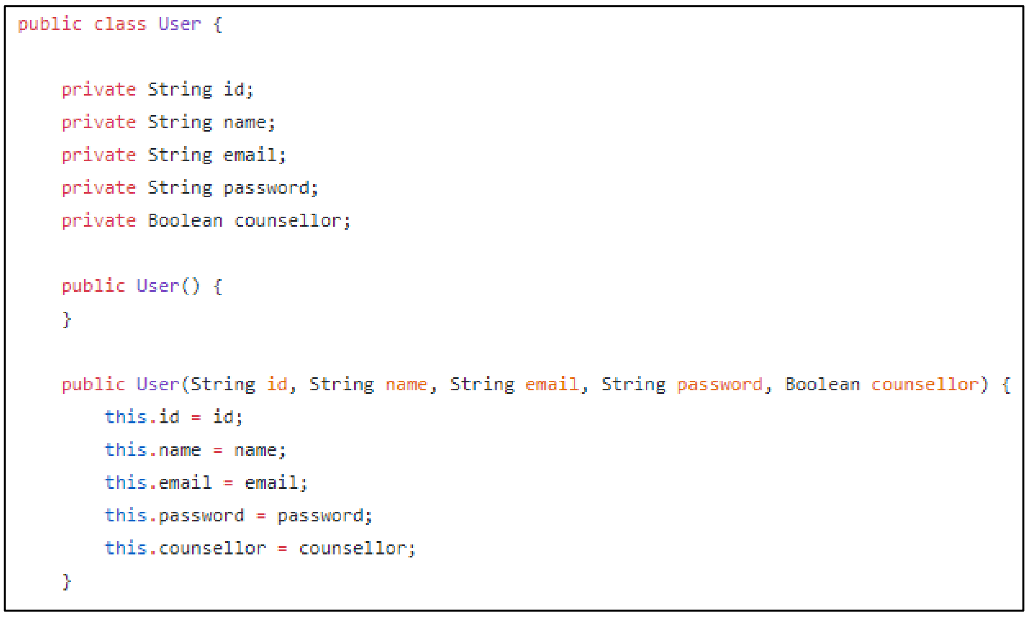
The middle layer (or middle tier) is the connection between the application interface and the system database. The middle layer development process includes creating Java classes and Firebase queries to insert and retrieve data from the database. The data inserted into the remote database includes questionnaires, sensed anxiety and user data. This section will discuss the creation the middle tier for the developed system.

4.5.2 Java Classes

Java Objects

The point of using Java object classes is to abstract the process of retrieving data from Firebase database. As mentioned, the Firebase database requires the use of objects to set and get node values. The means the following classes need to be created to allow interaction with the database:

User, Questionnaire, SensedAnxiety. All classes have a constructor and getters. When development first began, a constructor was not used in the classes. Instead, the classes used getters and setters to access data, which worked fine for pushing data to the cloud storage. When retrieving data from the database however not using a constructor caused problems. The object would always return null. On further research, it was realised that the object used with DataSnapshot.getValues() needs a default constructor. The following code fragment shows part of how the User object is constructed.

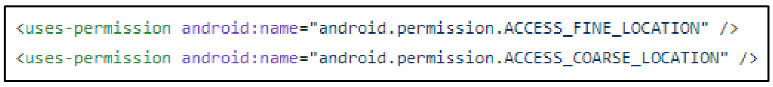


```java
public class User {

    private String id;
    private String name;
    private String email;
    private String password;
    private Boolean counsellor;

    public User() {
    }

    public User(String id, String name, String email, String password, Boolean counsellor) {
        this.id = id;
        this.name = name;
        this.email = email;
        this.password = password;
        this.counsellor = counsellor;
    }
```

Java Location

The initial plan for the developed system was that when an anxiety attack was sensed by the sensors, the application will capture the location of the user. As mentioned, the purpose of capturing the user's location is to allow them to become mindful of the environment of their anxiety.

Android devices have the ability to retrieve information about the user's current or last known location using the LocationManager. Using Google Play service location APIs an android application can request the last known location of a user's device. [65] This location is generally the user's current location. Android applications that require location knowledge about the device must request location permissions, shown below. "Android offers two location permissions: ACCESS_COARSE_LOCATION and ACCESS_FINE_LOCATION." [66] They differ by accuracy. ACCESS_COARSE_LOCATION returns a location within an accuracy equivalent to a city block and ACCESS_FINE_LOCATION returns a location within a more accurate value.



```xml
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

This project uses getLastKnownLocation method to retrieve the last known location, as mentioned it is generally the user's current position or close to that. The code to retrieve the user's location is shown below.

```
private Location getLastKnownLocation() {

    mLocationManager = (LocationManager) getApplicationContext().getSystemService(LOCATION_SERVICE);

    List<String> providers = mLocationManager.getProviders(true);
    Location bestLocation = null;

    for (String provider : providers) {

        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != Pack
            ActivityCompat.requestPermissions(this, new String[] {
                    Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.ACCESS_COARSE_LOCATION }, 10);
        }

        Location l = mLocationManager.getLastKnownLocation(provider);

        if (l == null) {
            continue;
        }
        if (bestLocation == null || l.getAccuracy() < bestLocation.getAccuracy()) {
            bestLocation = l;
        }
    }
    return bestLocation;
}
```

Once the location is captured, it needs to be parsed into a meaningful format before it is pushed to the cloud. The location object contains the longitude and latitude of the user's location. This latitude and longitude coordinates can be used to get the street name using the geocoder.getFromLocation method [67]. The street name will mean more to the user than coordinates.

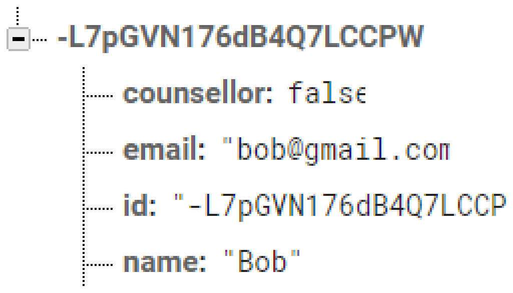
```
longitude = location.getLongitude();
latitude = location.getLatitude();
```

```java
private String location() throws IOException {

    Geocoder geocoder;
    List<Address> addresses;
    geocoder = new Geocoder(this, Locale.getDefault());

    addresses = geocoder.getFromLocation(latitude, longitude, 1);

    Address address = addresses.get(0);
    String streetName = address.getThoroughfare();

    if (streetName == null) {
        streetName = "Location unavailable";
    }

    return streetName;

}
```

Firebase Operations

As previously mentioned, objects need to be used when inserting into a Firebase database. As well as that, the database and node need to be identified before the application can attempt to insert data. As stated, if a referenced node does not exist Firebase will automatically create it. To identify the child inside the node, a unique key can be generated by Firebase. Then to insert the data, a new object must be instantiated with the values desired to be pushed to the cloud. The object is then passed to the setValue method, as demonstrated in the code below, and the values get committed to the node. The inserted data in the database can be view in Figure 42.

```java
String user_id = UsersDB.push().getKey();

UserDao user = new UserDao(user_id, userName, userEmail, password, false);

UsersDB.child(user_id).setValue(user);
```



```
-L7pGVN176dB4Q7LCCPW
    counsellor: false
    email: "bob@gmail.com
    id: "-L7pGVN176dB4Q7LCCP
    name: "Bob"
```

Figure 42 - Example of User data in Firebase database

When retrieving data from a Firebase database, the JSON structure needs to be taken into account. Meaning that nodes and children need to be iterated and ordered to achieve the desired results. As well as that, because of the JSON structure, joins do not exist in Firebase queries. Instead, queries can be nested and references to nodes can be created from pulled data.

Some issues were encountered with import references for gathering data from the Firebase database. The dependency added to the application was 'com.google.firebase'. The method used for extracting data from the Firebase database is the onDataChange method, this gets a DataSnapshot object from the Firebase database. This DataSnapshot requires a reference in Java. The reference used must match the dependency added in the build.gradle. When first trying to get the data snapshot from the database, the wrong import statement was used and went unnoticed for a while. During that time it caused the application to crash anytime it tried to access the data snapshot object.

The following examples indicate how data is queried from Firebase to retrieve the associated questionnaire for a selected anxiety episode. Firstly, the reference to the sensed_anxiety node and the children equal to user_id and location from the selected anxiety episode are established.

```
final DatabaseReference sensedDB = FirebaseDatabase.getInstance().getReference();
Query userRef = sensedDB.child("sensed_anxiety").orderByChild("user_ID").equalTo(user_id);
final Query locationRef = sensedDB.child("sensed_anxiety").orderByChild("location").equalTo(location);
```

An addListenerForSingleValueEvent is used to retrieve the data only once, as this use case requires the data to remain static. Then the onDataChange method is used to get a DataSnapshot back from a query.

```
userRef.addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        if(dataSnapshot.getValue() != null){

            locationRef.addListenerForSingleValueEvent(new ValueEventListener() {

                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {

                    if (dataSnapshot.getValue() != null) {

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

                            SensedAnxietyDao episode = episodeData.getValue(SensedAnxietyDao.class);
                            sensedID = episode.getSensedID();
                            final String loc = episode.getLocation();
```

Once the desired episode is found a new object can be instantiated with the data from the Firebase database. The sensed id of the episode can be used to find the questionnaire node with that id from the new object. Once the data is obtained it can be used to populate views.

Another challenge that was encountered during development was trying to use instantiated objects with Firebase data, outside the onDataChange method. When an object is populated with values inside the onDataChange method, it only has those values inside that method. This means that populating views with data from a Firebase node needs to be carried out inside that method. During development, this altered how some activities were developed.

4.5.3 Conclusion

The middle layer code allows the application interface and databases to communicate. This is achieved using Java objects and classes. The objects and classes let data be accessed and populated in interface views. Despite some issues with dependencies and instantiated objects from Firebase, the development of the middle layer brought together the back-end and front-end.

4.6 Final Classes

The final classes for the Anxiety Manager system are as follow:

Actvities	*Fragments*	*Objects*	*SQL*	*Adapters*
CalendarActivity	LocationFragment	Questionnaire	DatabaseManager	EmotionDialogAdapter
ChoosenPatientActivity	ReactionFragment	SensedAnxiety		ThoughtsAdapter
CounsellorHomeActivity	SubjectFragment	User		SubjectAdapter
HomeActivity				
LoginActivity				
EmotionActivity				
PhysicalActivity				
RateOneMoodActivity				
RateTwoMoodActivity				
ReactActvity				
RegisterActivity				
SensedActivity				
ThoughtsActivity				
SubjectActvity				

4.7 Overall Conclusion

Overall, developing the system was a learning experience with some challenges. Despite considering an intricate design and predicting risks, unforeseen problems were encountered and decisions needed to be made at once about workarounds. Workarounds were determined if they proved the project idea was feasible. Challenges ranged from substantial issues like Arduino and Android connection incompatibility, to more trivial concerns such as incorrect import statements and constructors. Some successfully developed features include the application interface and the system back-end. Once features were developed, testing of them was carried out. The following chapter looks at how testing was used to devise a robust system.

Chapter 5 System Validation

5.1 Introduction

When a system is created, it needs to be tested and evaluated to catch bugs and problems missed during development and design phases. Some of this testing occurred parallel to development, while others happened after the whole system development. As well as testing, evaluation needs to be carried out to ensure the system is usable. Given the user focus of the project, the system validation is an essential phase of the project lifecycle.

5.2 Testing

Having designed and developed a system for this project, creating tests that are just as robust as the system is vital to its success. As the project has a user-centred approach, having a user centred testing is important. White Box testing was planned for the system but it was considered too time-consuming. More of an emphasis needed to be placed on user focus testing. The final testing that occurred for the developed system consists of usability testing, requirement-based testing and sensor testing.

5.2.1 Usability Testing

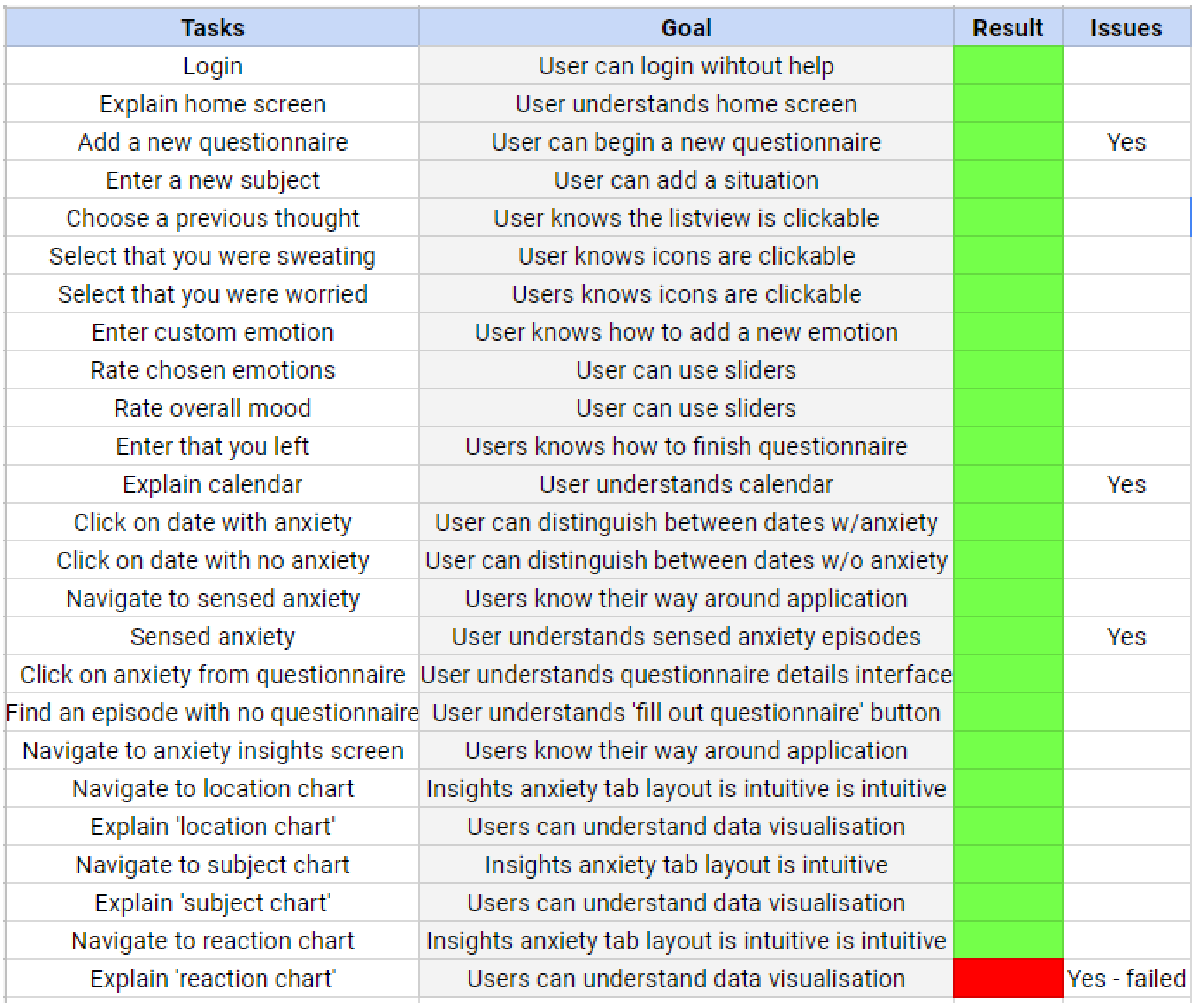
Usability testing is the most important test for this system. The objective of this project is to create an easy-to-use system for users of all IT levels. As well as that, Anxiety Manager aims to give users the first step in combating their anxiety with tracking. This system must be tested to ensure that objective is met. Individual feedback was collected during design phases and the same users were asked to complete usability tests. End users varied from users who have anxiety to users who work to treat individuals with anxiety and have experience in anxiety tracking. The usability tests consisted of giving users tasks to complete using the Android application and the tester watching how easily a user carried out that task. Before tests took place, the outcomes and script for the test were decided on. The test script can be found here.

The goals of usability testing are as follows:

- Can users easily use the application?
- Can users log a questionnaire successfully?
- Can users navigate through the application without confusion?
- Can users recognize the language within the application?
- Are users overwhelmed by using the application?
- Where are users stumbling?

Results

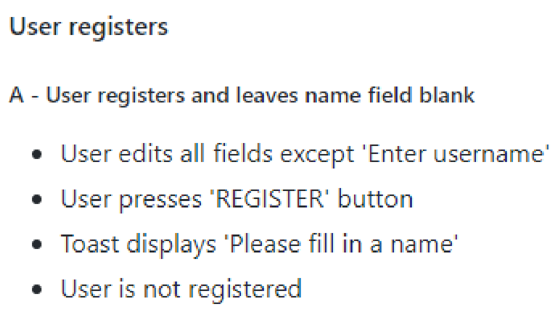
The overall results and feedback from the usability tests indicated that some text in the application needed to be more explicit what was required of the user. The general feedback from the usability tests was that the application was easy to use and straightforward. Users were never overwhelmed by the application by errors or features which was the primary goal of designs. Feedback from users during usability tests could also be used for future work and designs. The below chart shows an example of one usability test went. The user feedback was taken and some small design changes were made based on it (such as button text, icon sizes).



Tasks	Goal	Result	Issues
Login	User can login wihtout help	🟩	
Explain home screen	User understands home screen	🟩	
Add a new questionnaire	User can begin a new questionnaire	🟩	Yes
Enter a new subject	User can add a situation	🟩	
Choose a previous thought	User knows the listview is clickable	🟩	
Select that you were sweating	User knows icons are clickable	🟩	
Select that you were worried	Users knows icons are clickable	🟩	
Enter custom emotion	User knows how to add a new emotion	🟩	
Rate chosen emotions	User can use sliders	🟩	
Rate overall mood	User can use sliders	🟩	
Enter that you left	Users knows how to finish questionnaire	🟩	
Explain calendar	User understands calendar	🟩	Yes
Click on date with anxiety	User can distinguish between dates w/anxiety	🟩	
Click on date with no anxiety	User can distinguish between dates w/o anxiety	🟩	
Navigate to sensed anxiety	Users know their way around application	🟩	
Sensed anxiety	User understands sensed anxiety episodes	🟩	Yes
Click on anxiety from questionnaire	User understands questionnaire details interface	🟩	
Find an episode with no questionnaire	User understands 'fill out questionnaire' button	🟩	
Navigate to anxiety insights screen	Users know their way around application	🟩	
Navigate to location chart	Insights anxiety tab layout is intuitive is intuitive	🟩	
Explain 'location chart'	Users can understand data visualisation	🟩	
Navigate to subject chart	Insights anxiety tab layout is intuitive	🟩	
Explain 'subject chart'	Users can understand data visualisation	🟩	
Navigate to reaction chart	Insights anxiety tab layout is intuitive is intuitive	🟩	
Explain 'reaction chart'	Users can understand data visualisation	🟥	Yes - failed

5.2.2 Requirement-Based Testing

The purpose of the requirement based testing for this developed system is to ensure that designs were followed closely and that user requirements were met. This testing was carried out with simple test-cases that were formulated along the development process. The testing was carried out on two Android devices, one was running Android 4.4.4 and the other was running 6.0.1. This ensured that the application was consist across different screen sizes and different Android versions. Each feature was tested once it was developed. The requirements-based testing mostly tests that UI is correct and that the data being pushed to the cloud is as inputted by the user. The complete version of test-cases can be found here. All tests passed the requirement testing, but some bugs were encountered as the application was being tested during development. A snippet of a test-case for the register feature can be found below:



User registers

A - User registers and leaves name field blank

- User edits all fields except 'Enter username'
- User presses 'REGISTER' button
- Toast displays 'Please fill in a name'
- User is not registered

Results

The requirements-based testing allowed bugs to be caught in the software. Most remaining bugs in the system were UI based. Such as incorrect text, incorrect colours or incorrect transitions. Once bugs were fixed per feature, the feature was validated again using the test-cases. When a test-case was passed, it was said that the feature was complete.

5.2.3 Sensor Testing

When the Arduino and sensors were being configured, they needed to be tested and calibrated. The sensors need to be tested while the user is resting and while they are under "anxiety". In order for a state of anxiety to mocked, users were asked to watch horror film trailers while wearing the sensors, and the serial plotter was monitored for changes. Unfortunately, the pulse sensor could not be tested as the configuration failed. The grove sensor was used during sensor testing.

Results

Serial plotter while the user was resting shown below in Figure 43. The resistance while resting for this user varied between 260 - 200, which is not a large difference. As well as that the values are consist between 260 and 200 which is a indiciator the user is relaxed.

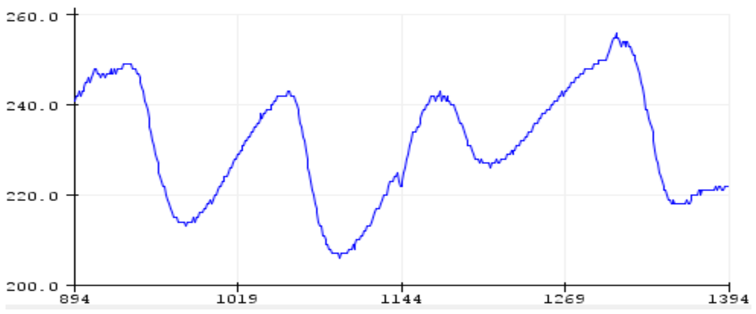


Figure 43 - Serial plotter for the user when resting

Serial plotter while the user was watching a horror movie trailer can be viewed in Figure 44. From the serial plotter we can see firstly, the user appears to be quite calm and the values are normal. When the jump scares happen in the trailer, huge drops in the resistance can be seen as the user. The range of the resistance was in the middle of 280 and 200. The times when the user felt stressed can be seen between 5515 seconds and 5615 seconds and again between 5715 and 5815.

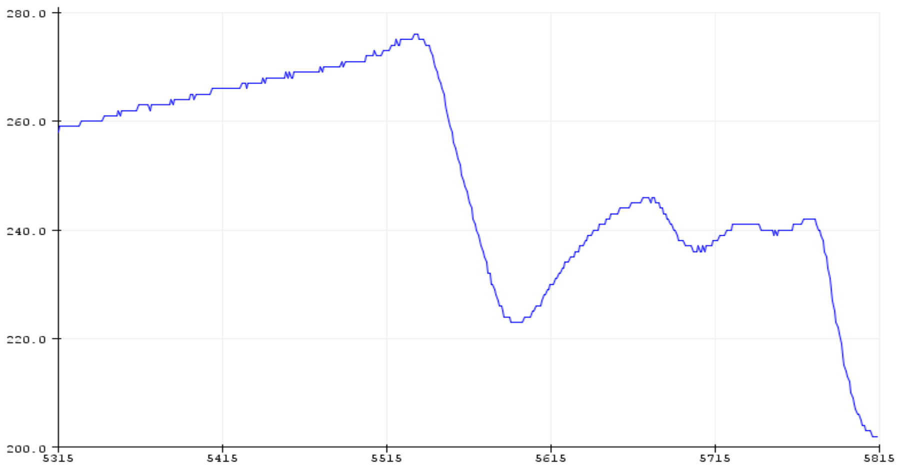


Figure 44 - Serial plotter for the user when watching a scary movie trailer

By carrying out sensor tests, it could be established how the sensors work. If there were no issues with connecting the Arduino and Android device further sensor testing would have been used to discover the threshold to determine when anxiety attacks occur.

5.3 Evaluation

This project focuses heavily on the user's interaction with the system and the accessability of the system. Following testing, which indicated that the system worked for users, the appliaciton needs to be evaluated. This chapter will evaluate the applciation usability. It will also outline the overall advantages and disadvantages of the system.

5.3.1 Usability

Prior to developing the system, during the research phase, existing anxiety applications were evaluated using Nielsen's Heuristics. The outcome of that evaluation was that an anxiety-focused-system must keep users informed, speak their language, have good error prevention, promote recognition rather than recall and provide clear documentation. These key heuristics help prevent users from becoming even more overwhelmed while logging their anxiety attacks, which is something this project focuses on. The system was evaluated under those key heuristics discovered in Section 2.3.2.

Speak the user's language by using real-world examples and non-technical terms.

The developed application does speak the user's language and emulate real-world examples. The questionnaire uses colloquial and casual terms to ask the user about their anxiety attack, such as "What's up" and "I stook around". As well as that, icons and emojis are present in the application and designed in a canonical perspective to match their real-world objects [64].

Have error prevention for users by including text input constraints and delete action constraints.

The developed system has good error prevention. Any input required by a user must be filled and users can go back and change their response to a questionnaire.

Avoid users recall terms, icons or actions across the app, but instead invite users to recall them.

The application uses the same icons, text, terms and actions across it. Users are not required to retain any information from one place in the application to another.

Provide users with clear documentation for each activity with minimal text.

This is something the developed system lacks. Unfortunately, time ran out during development and documentation could not be placed within the application. The intention was to have an icon on every screen that explained what was required of the user on each screen.The application interactions are quite minimal. This reduces the need for a lot of documentation but to further the accessibility of the application for all types of users documentation needs to be added in the future.

Have user control and freedom for every action to avoid user's making wrong decisions.

All actions within the application can be undone. The system, however, is missing the freedom to delete or achieve submitted questionnaires and anxiety attacks. This is another heuristic that needs to be improved in future work.

Keep users informed throughout the application labels and titles.

All activities in the developed system keep the users informed using a title on the toolbar. As well as that, any icons in the application are accompanied by a label to avoid misinterpretation.

5.3.2 Project Analysis

A demonstration of the developed applcation can be found here.

It can be said Anxiety Manager possesses the following advantages:

- Tackles the problem with outdated traditional anxiety tracking solutions.
- Prevents individuals running the risk of losing their anxiety tracking information because of the cloud-based technology.
- Has a minimal design that promotes to calm its users.
- Personal information is secure within the application.
- Works offline.
- Gives users a way to visualise their anxiety attacks.

It can be said Anxiety Manager possess the following disadvantages:

- Currently counsellors can access any patients data who is registered with the app.
- Poor documentation.
- Not appropriate for every individual with anxiety. Some people may simply prefer pen and paper tracking.

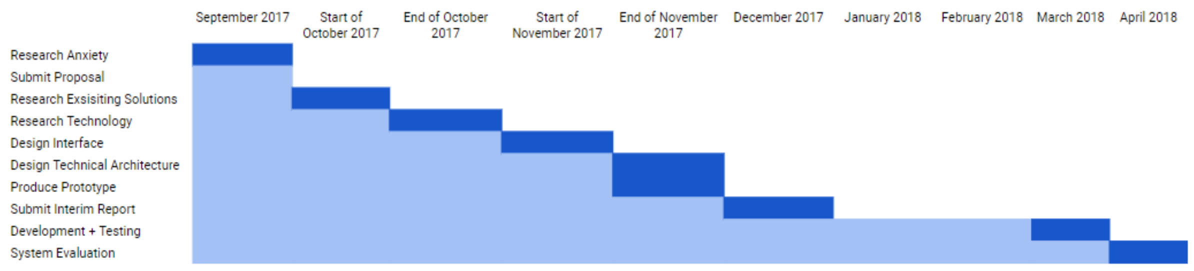
5.3 Overall Validation

Overall, from testing, usability evaluation and analysis. The testing results determined that the application works well and the evaluation concluded that the application is usable by a range of users. Users noted during testing, that the application was easy to use and didn't cause any more anxiety. From NH, it can be said the application has good error prevention and complies with real-world conventions. Although after analysis, it was recorded that the application lacks documentation, freedom and the accurate counsellor profiles. However, these can be addressed, in future work.

Chapter 6 Project Plan

This chapter will discuss the plan created for the project and the priority given to features for development. As well as that, the chapter explores the future of Anxiety Manager.

6.1 Project Plan



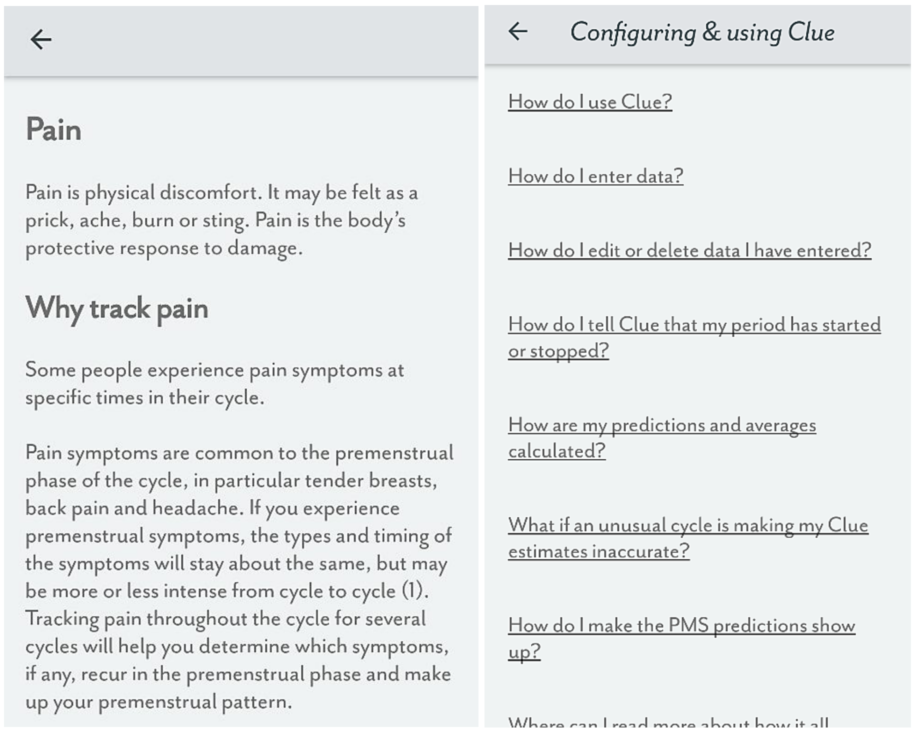
The project plan was to firstly research and design the whole system, and subsequently develop, test and fix features one by one. This approach worked quite well but only because of the chosen methodology. Overall the Agile methodology worked favourably for this project. It permitted flexibility when issues, new decisions and "dead-ends" were encountered. Agile allowed the design, development and testing of the project to happen in iterations. The project timeline can be viewed in the Gantt chart above.

The initial feature priority was given: Homepage, Login/Register, Profiles, Sensed Anxiety, Questionnaire, Insights and Calendar. When the project objective and designs were revisited before development, it was realised that this priority was all wrong. There needed to be a preference given to features that allowed users to track their anxiety attacks and view the insights of those attacks. The final feature priority was Questionnaire, Sensed Anxiety, Sensing Anxiety List, User Profile, Calendar, Anxiety Insights, Home Page, and Counsellor Profile. Because of trivial issues with development and problems with configuring sensors, the project plan was not followed completely. As a result, the counsellor profile could not be developed fully in time.

6.2 Future Work

Anxiety Manager has potential to expand and improve in the future. Firstly, one feature that will be improved eventually is the counsellor profile, as presently this feature doesn't work as planned. The counsellor profile needs to be more rigid in the future and only allow counsellors who have permission from their patients to see their profile. A possibility for users with anxiety and their counsellors to add one another to the profile should be developed and currently, that is something the application lacks really. The project objective is met with the system developed as is. Users can gain insights into their anxiety regardless if their counsellor is involved or not. However, an extra person to discuss those insights with is what will enable an individual to overcome their issues.

As well as that, future work needs to be carried out to improve the sensor issue. The workaround works well for the project demonstration, but for this system to benefit a person with an anxiety, a device to sense anxiety needs to be introduced. Integrating the application with a Fitbit or other smartwatches in the future would really improve the practicality of Anxiety Manager. Wearable devices would match how people are monitoring themselves currently in real life situations.


Pain

Pain is physical discomfort. It may be felt as a prick, ache, burn or sting. Pain is the body's protective response to damage.

Why track pain

Some people experience pain symptoms at specific times in their cycle.

Pain symptoms are common to the premenstrual phase of the cycle, in particular tender breasts, back pain and headache. If you experience premenstrual symptoms, the types and timing of the symptoms will stay about the same, but may be more or less intense from cycle to cycle (1). Tracking pain throughout the cycle for several cycles will help you determine which symptoms, if any, recur in the premenstrual phase and make up your premenstrual pattern.

Configuring & using Clue

How do I use Clue?

How do I enter data?

How do I edit or delete data I have entered?

How do I tell Clue that my period has started or stopped?

How are my predictions and averages calculated?

What if an unusual cycle is making my Clue estimates inaccurate?

How do I make the PMS predictions show up?

Figure 45 - Clue app documentation [68]

The final improvement to the application that needs to be developed is proper documentation. The app was developed to be easy-to-use and simple. To ensure every user has an equal opportunity to use this application to overcome their anxiety, documentation needs to be provided on how this application works. The application 'Clue', a female health mobile application similar to Anxiety Manager, invites users to track symptoms about their health. Clue provides its users with extensive documentation about why they should track certain symptoms during their menstrual cycle, Figure 45. As well as that, Clue doesn't expect all its users have access to professional healthcare and aims to answer any questions users may have about their cycle. The way this application handles documentation should be looked at when the documentation for Anxiety Manager is being produced.

Other future feature work could implement more innovative technology such as machine learning or systems intelligence. In the future, the application could be used to predict anxiety attacks based on the user's inputs and usual anxiety attacks. For example, a user appears to be approaching a congested area. The system could warn them about potential attacks if their anxiety triggers include crowded areas or places with lots of people. Similarly, the anxiety calendar could be expanded to integrate the user's personal calendar. The system could make users aware of life events that may be triggering anxiety, like tests or anniversaries. Overall any future work should aim to give users as much information about their anxiety as possible.

Chapter 7 Conclusion

This chapter reflects on the project. Including how the project objective was met, what would be changed if the project was carried out again and what experience can be taken from the project.

The objective of this project was to develop a non-invasive, easy to use, mobile system that provides understanding to a user about their anxiety and, overcome the issues associated with current tracking approaches. This objective was met by developing an Android solution that provides a relaxed way to track anxiety. Inspiration for this project was drawn from the argument that technology is having a negative effect on people's mental health. It is accurate that there has been an increase in individuals battling mental health issues, since the development of mobile technology. But, it is not a viable solution to try and banish technology from people's lives. The hope was to benefit from technology already present in people's lives (such as mobile phones and smart wearables) and build a solution for people to take the first steps in understanding their own anxiety disorder. This project considers the development of a proof of concept application that demonstrates that mobile technology can replace out of date pen and paper tracking methods. As well as that, this project offers more options missing in traditional solutions. Pen and paper tracking cannot provide visual insights into anxiety habits and environments or, counsellor access to patient anxiety attacks like Anxiety Manager can. This proof of concept determines a comforting way for individuals to track anxiety without added stress.

If the project was undertaken again, there would be a quicker response to change. Although the Agile method was used for the project lifecycle, at times change was uncomfortable and unwanted. A lot of time was wasted trying to prevent change from happening, i.e. spending over the allocated time getting sensors to work with the Android device, instead of just accepting the requirement needed to be adjusted to fit the project environment. With this, crucial features like the counsellor profile were missed out on. Although, being placed in uncomfortable positions (like technology not functioning as intended or plans that weren't as rigid as perceived) has strengthened the knowledge gained from carrying out this project. The lifespan of this project mimics that of an industry project. The biggest experiences learned from this project are the ability to critique own personal work, the ability to adapt to changing scope and the ability to move on from mistakes. These skills will be needed again following this degree in industry. As well as that, now knowing how to enhance development work with research, design and validation is invaluable. It can be said that the lifespan of this project provided an invaluable amount of knowledge and perspective to the years spent studying computer science.

Chapter 8 Bibliography

1. Mental Health Ireland. (2018). [online] Available at: http://www.mentalhealthireland.ie/a-to-z/anxiety/ [Accessed 26 Mar. 2018].
2. anxietycentre.com. (2018). Anxiety: what it is and what causes it. [online] Available at: http://www.anxietycentre.com/anxiety.shtml [Accessed 26 Mar. 2018].
3. Merriam-webster.com. (2018). Definition of ANXIETY. [online] Available at: https://www.merriam-webster.com/dictionary/anxiety [Accessed 26 Mar. 2018].
4. Bame, M. (2011). 6 Facts about Public Speaking Anxiety. [online] Breaking Down Barriers. Available at: http://bdbcommunication.com/6-facts-about-public-speaking-anxiety/ [Accessed 26 Mar. 2018].
5. AnxietyBC. (2017). Anxiety 101. [online] Available at: https://www.anxietybc.com/adults/anxiety-101 [Accessed 26 Mar. 2018].
6. Bolger, C. (2017). Interview. A discussion about anxiety, the physical symptoms of anxiety and how to track anxiety. Transcript available at: https://drive.google.com/file/d/1fFPl5LNbv-kbuiZu5c5o5tt141C9eFt3/view?usp=sharing
7. Medical News Today. (2017). Anxiety: Causes, Symptoms and Treatments. [online] Available at: https://www.medicalnewstoday.com/info/anxiety [Accessed 26 Mar. 2018].
8. Hellolife.net. (2018). Anxiety Disorder: Who Does It Affect?. [online] Available at: https://www.hellolife.net/anxiety/b/anxiety-disorder-who-does-it-affect [Accessed 27 Mar. 2018].
9. Calmclinic.com. (2018). What Happens During an Anxiety Attack?. [online] Available at: https://www.calmclinic.com/anxiety/attacks/what-happens-during-one [Accessed 27 Mar. 2018].
10. wikiHow. (2018). How to Track Your Anxiety. [online] Available at: https://www.wikihow.com/Track-Your-Anxiety# [Accessed 27 Mar. 2018].
11. Verywell Mind. (2018). How to Use a Panic Attack Diary. [online] Available at: https://www.verywellmind.com/anxiety-and-panic-attack-diary-2584057 [Accessed 27 Mar. 2018].
12. Psychology Today. (2018). 5 Ways to Stop Your Racing Thoughts. [online] Available at: https://www.psychologytoday.com/us/blog/women-s-mental-health-matters/201604/5-ways-stop-your-racing-thoughts [Accessed 27 Mar. 2018].
13. Calmclinic.com. (2018). Anxiety and Mood Swings. [online] Available at: https://www.calmclinic.com/anxiety/symptoms/mood-swings [Accessed 27 Mar. 2018].
14. Calmclinic.com. (2018). How Are Hormones And Anxiety Related?. [online] Available at: https://www.calmclinic.com/anxiety/causes/hormones [Accessed 27 Mar. 2018].
15. Study.com. (2018). Robert Plutchik's Wheel of Emotions - Video & Lesson Transcript | Study.com. [online] Available at: https://study.com/academy/lesson/robert-plutchiks-wheel-of-emotions-lesson-quiz.html [Accessed 27 Mar. 2018].
16. Donaldson, M. (2017). Plutchik's Wheel of Emotions - 2017 Update • Six Seconds. [online] Six Seconds. Available at: http://www.6seconds.org/2017/04/27/plutchiks-model-of-emotions/ [Accessed 27 Mar. 2018].
17. Psychology Today. (2018). Avoidance of Anxiety as Self-Sabotage: How Running Away Can Bite You in the Behind. [online] Available at: https://www.psychologytoday.com/us/blog/overcoming-self-

sabotage/201005/avoidance-anxiety-self-sabotage-how-running-away-can-bite-you [Accessed 27 Mar. 2018].

18. Flourishnthrive.files.wordpress.com. (2018). [online] Available at: https://flourishnthrive.files.wordpress.com/2012/06/dailymood.jpg [Accessed 9 Apr. 2018].
19. Gibbs, N. (2018). Your Life Is Fully Mobile | TIME.com. [online] TIME.com. Available at: http://techland.time.com/2012/08/16/your-life-is-fully-mobile/ [Accessed 27 Mar. 2018].
20. Karen Wall, B. (2018). The Worry Box Mobile App - Behavioral App Reviews. [online] Behavioral App Reviews. Available at: http://telehealth.org/apps/behavioral/the-worry-box-mobile-app [Accessed 28 Mar. 2018].
21. Play.google.com. (2018). [online] Available at: https://play.google.com/store/apps/details?id=com.uwe.myoxygen&hl=en [Accessed 28 Mar. 2018].
22. App Store. (2016). What's Up? - A Mental Health App on the App Store. [online] Available at: https://itunes.apple.com/us/app/whats-up-a-mental-health-app/id968251160?mt=8 [Accessed 28 Mar. 2018].
23. Nielsen Norman Group. (2018). 10 Heuristics for User Interface Design: Article by Jakob Nielsen. [online] Available at: https://www.nngroup.com/articles/ten-usability-heuristics/ [Accessed 28 Mar. 2018].
24. Costa, A. (2017). What are Emoji's? How and When to Use Them. [online] groovyPost. Available at: https://www.groovypost.com/howto/what-are-emojis-how-and-when-to-use-them [Accessed 21 Oct. 2017].
25. McGinn, D. (2017). Say it with a smiley face: The emotional work of emojis. [online] The Globe and Mail. Available at: https://beta.theglobeandmail.com/life/relationships/say-it-with-a-smiley-face-the-emotional-work-of-emojis/article34930253/?ref=http://www.theglobeandmail.com& [Accessed 21 Oct. 2017].
26. Thomas, D. (2017). Get the iPhone's Emoji on Your Google Pixel or Pixel XL. [online] WonderHowTo. Available at: https://android.gadgethacks.com/how-to/get-iphones-emoji-your-google-pixel-pixel-xl-0176052/ [Accessed 24 Nov. 2017].
27. Social. (2017). 7 Science-Based Reasons to Use Emoticons. [online] Available at: https://blog.bufferapp.com/7-reasons-use-emoticons-writing-social-media-according-science [Accessed 21 Oct. 2017].
28. Taylor & Francis. (2017). Emoticons in mind: An event-related potential study. [online] Available at: http://www.tandfonline.com/doi/abs/10.1080/17470919.2013.873737?journalCode=psns20#.VLadY2TF871 [Accessed 21 Oct. 2017].
29. Harley, A. (2017). Icon Usability. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/icon-usability/ [Accessed 22 Nov. 2017].
30. Learn. (2017). Infographic Design - Tips and Inspiration By Canva. [online] Available at: https://www.canva.com/learn/how-to-design-infographics/ [Accessed 22 Nov. 2017].
31. Nces.ed.gov. (2017). How Do I Choose Which Type of Graph to Use?-NCES Kids' Zone. [online] Available at: https://nces.ed.gov/nceskids/help/user_guide/graph/whentouse.asp [Accessed 22 Nov. 2017].
32. Community.fitbit.com. (2016). Fitbit tracks anxiety as an activity?. [online] Available at: https://community.fitbit.com/t5/Charge-HR/Fitbit-tracks-anxiety-as-an-activity/td-p/1554029?nobounce [Accessed 29 Mar. 2018].

33. BuzzFeed. (2018). I Used A Fitbit To Try To Conquer My Stress. [online] Available at: https://www.buzzfeed.com/beckybarnicoat/could-a-fitbit-help-me-conquer-my-stress?utm_term=.ey4m6NyqP#.cin9Rn6dz [Accessed 29 Mar. 2018].
34. Allpsychologycareers.com. (2018). What is anxiety counseling?. [online] Available at: http://www.allpsychologycareers.com/topics/anxiety-counseling.html [Accessed 29 Mar. 2018].
35. Lifewire. (2018). What Is a Mobile Operating System?. [online] Available at: https://www.lifewire.com/what-is-a-mobile-operating-system-2373340 [Accessed 29 Mar. 2018].
36. Android, t. (2017). Android, the world's most popular mobile platform | Android Developers. [online] Developer.android.com. Available at: https://developer.android.com/about/android.html [Accessed 20 Nov. 2017].
37. Macrumors.com. (2018). Android Continues to Have More Loyal Customers Than iOS. [online] Available at: https://www.macrumors.com/2018/03/08/android-ios-customer-loyalty/ [Accessed 29 Mar. 2018].
38. Developer.android.com. (2018). Location & Context | Android Developers . [online] Available at: https://developer.android.com/training/location/index.html [Accessed 29 Mar. 2018].
39. dummies. (2018). Why Develop iOS Applications? - dummies. [online] Available at: http://www.dummies.com/web-design-development/mobile-apps/why-develop-ios-applications/ [Accessed 29 Mar. 2018].
40. Developer.apple.com. (2018). Getting the User's Location. [online] Available at: https://developer.apple.com/library/content/documentation/UserExperience/Conceptual/LocationAwarenessPG/CoreLocation/CoreLocation.html [Accessed 29 Mar. 2018].
41. Grove - GSR. (2015). 1st ed. [PDF] Seeed Studio. Available at: http://www.mouser.com/catalog/specsheets/Seeed_101020052.pdf [Accessed 18 Oct. 2017].
42. ElProCus - Electronic Projects for Engineering Students. (2017). Heart Beat Sensor - How to Measure Heart Beat: Working and Application. [online] Available at: https://www.elprocus.com/heartbeat-sensor-working-application/ [Accessed 18 Oct. 2017].
43. Instructables.com. (2017). Basic Belt Respiration Sensor. [online] Available at: http://www.instructables.com/id/Quick-and-dirty-Respiration-Sensor/ [Accessed 18 Oct. 2017].
44. Cleveland Clinic. (2017). Vital Signs | Vital signs are used to measure the body's basic functions. [online] Available at: https://my.clevelandclinic.org/health/articles/vital-signs [Accessed 22 Nov. 2017].
45. Soil Moisture Sensor. (2010). [PDF] Vernier Software & Technology, p.1. Available at: http://www.tvdsb.ca/uploads/ScienceProbeware/soilmoisture.pdf [Accessed 18 Oct. 2017].
46. Di Justo, P. (2017). Raspberry Pi or Arduino? One Simple Rule to Choose the Right Board [online] Make: DIY Projects and Ideas for Makers. Available at: https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/ [Accessed 22 Nov. 2017].
47. www.tutorialspoint.com. (2018). Android SQLite Database. [online] Available at: https://www.tutorialspoint.com/android/android_sqlite_database.htm [Accessed 29 Mar. 2018].

48. File?, W. (2018). What is the advantage of Using SQLite rather than File?. [online] Stackoverflow.com. Available at: https://stackoverflow.com/questions/19946298/what-is-the-advantage-of-using-sqlite-rather-than-file [Accessed 29 Mar. 2018].
49. Firebase, C. (2018). Connect to Firebase | Android Studio . [online] Developer.android.com. Available at: https://developer.android.com/studio/write/firebase.html [Accessed 29 Mar. 2018].
50. AndroidPub. (2017). Why we used firebase? – AndroidPub. [online] Available at: https://android.jlelse.eu/why-we-used-firebase-12c0cb6f0150 [Accessed 29 Mar. 2018].
51. Allpsychologycareers.com. (2017). What is anxiety counseling?. [online] Available at: http://www.allpsychologycareers.com/topics/anxiety-counseling.html [Accessed 21 Oct. 2017].
52. Base36.com. (2017). Agile & Waterfall Methodologies – A Side-By-Side Comparison | Base36. [online] Available at: http://www.base36.com/2012/12/agile-waterfall-methodologies-a-side-by-side-comparison/ [Accessed 5 Nov. 2017].
53. Naiman, L. (2017). Design Thinking as a Strategy for Innovation. [online] Creativity at Work. Available at: https://www.creativityatwork.com/design-thinking-strategy-for-innovation/ [Accessed 15 Nov. 2017].
54. Marvel Prototyping. (2017). Free mobile & web prototyping (iOS, Android) for designers – Marvel. [online] Available at: https://marvelapp.com/pop/ [Accessed 20 Nov. 2017].
55. GitHub. (2018). SundeepK/CompactCalendarView. [online] Available at: https://github.com/SundeepK/CompactCalendarView [Accessed 10 Apr. 2018].
56. GitHub. (2018). PhilJay/MPAndroidChart. [online] Available at: https://github.com/PhilJay/MPAndroidChart [Accessed 10 Apr. 2018].
57. Developer.android.com. (2018). AlarmManager | Android Developers . [online] Available at: https://developer.android.com/reference/android/app/AlarmManager.html [Accessed 10 Apr. 2018].
58. Wiki.seeedstudio.com. (2018). Grove - GSR Sensor. [online] Available at: http://wiki.seeedstudio.com/Grove-GSR_Sensor/ [Accessed 10 Apr. 2018].
59. Wiki.seeedstudio.com. (2018). Grove - GSR Sensor. [online] Available at: http://wiki.seeedstudio.com/Grove-GSR_Sensor/ [Accessed 10 Apr. 2018].
60. GitHub. (2018). OmarAflak/Arduino-Library. [online] Available at: https://github.com/OmarAflak/Arduino-Library [Accessed 10 Apr. 2018].
61. 'Cause You're Stuck. (2017). Connect Arduino to Android through USB - 'Cause You're Stuck. [online] Available at: https://causeyourestuck.io/2017/05/24/connect-android-arduino-usb/ [Accessed 10 Apr. 2018].
62. Hellman, E. (2014). Android programming. Chichester, West Sussex: Wiley, p.172.
63. Developer.android.com. (2018). ListAdapter | Android Developers . [online] Available at: https://developer.android.com/reference/android/widget/ListAdapter.html [Accessed 11 Apr. 2018].
64. Hellman, E. (2014). Android programming. Chichester, West Sussex: Wiley, p.86.
65. Developer.android.com. (2018). Getting the Last Known Location | Android Developers . [online] Available at: https://developer.android.com/training/location/retrieve-current.html [Accessed 11 Apr. 2018].
66. Google Developers. (2018). Location Data | Google Maps Android API | Google Developers. [online] Available at: https://developers.google.com/maps/documentation/android-api/location [Accessed 11 Apr. 2018].
67. Hellman, E. (2014). Android programming. Chichester, West Sussex: Wiley, p.258.

68. Helloclue.com. (2018). Clue: Period and Ovulation Tracker for iPhone and Android. [online] Available at: https://helloclue.com/ [Accessed 12 Apr. 2018].
69. Colour-affects.co.uk. (2018). *Psychological Properties Of Colours - Colour Affects*. [online] Available at: http://www.colour-affects.co.uk/psychological-properties-of-colours [Accessed 13 Apr. 2018].