UNIVERSITY OF BIRMINGHAM

School of Computer Science

Anxiet-ease

Sarina Saqib
2249047

MSc Computer Science

Supervised by: Ian Kenny & Mark Lee

September 2021

Abstract

This project aimed to develop a mobile application as a tool for psychological intervention. Access to psychological intervention remains to be an issue in even the most developed of communities, leaving sufferers of mental health conditions to look else where for help. There currently exists a big gap in the market of mobile applications for psychologically based apps that help ease symptoms of mental health issues, despite the growing evidence surrounding their potential. The Android application, Anxiet-ease, was developed to fill this gap and help ease users symptoms of anxiety. Anxiet-ease was developed using Java and Android studio, through a psychological lens utilising existing clinical research and practise. A range of activities have been designed and implemented, aiming to provide the user with an array of gaming distractions and mindfulness activities, allowing them to lead their own intervention. During user experience research, Anxiet-ease was found to excel at its intended use of lowering over all anxiety levels of users by an average of 3 points (on a 10 point scale). The majority of participants also stated they would use Anxiet-ease again, and recommend it to a friend or family member. With proof of concept, further expansion of this application into an iOS application has proved to be imperative.

Keywords: Java, Android, Android Studio, Mobile development, Mobile app, Anxiety, Intervention

Contents

List of Figures

1.0 Introduction

The aim of this project was to develop a novel mobile application, designed through a psychological lens, to help ease the symptoms of anxiety. The application, Anxiet-ease, has been developed in Java using Android studio. The activities within the application make use of existing intervention as well as newer research to create a range of distraction and mindfulness activities.

Currently, there exists limited options available for mental health intervention, and excessive barriers to intervention surround those that do. Research into the use of mobile applications for psychological intervention purposes, show promising results for those applications that are research based. This project intends to build on the work already accomplished in this field, building a psychologically sound mobile application to ease anxiety.

This project has a psychologically ambitious goal of utilising a concept called flow as a tool for anxiety intervention. Research in this field of application is currently very limited but extremely promising. The main outcome of flow is total absorption in an activity. Anxiet-ease intends to harness this into a distraction from anxiety through the use of easy to understand games.

There remains to be a distinct lack of choice when it comes to psychologically based applications on the market. Those that do exist have one activity on offer, and those that are psychologically researched are often expensive, have long registration processes, and a lot of data collection. Anxiet-ease intends to fill this gap in the market, offering users a variety of psychologically designed, novel activities. Allowing the user to guide their own intervention.

For this project to be deemed a psychological success, the majority of participants from the user research survey would have an overall psychologically positive experience of the application. The majority of the participants would also need to show a reduction in overall anxiety levels before and after using Anxiet-ease. This was measured on a ten point scale. Finally, the majority of participants must feel the distraction games were beneficial psychologically. Providing evidence for the success of flow being utilised.

For this project to be an engineering success, the application designed must be robust and modular. The application must have a smooth performance with no bugs or glitches during user testing. Good software engineering principles need to be followed through the project. The application must have high reliability and usability. The code behind the software must also be easily readable and well documented.

The aim of this report is to first detail the research and theory behind Anxiet-ease, including the psychological and computer science background necessary to fully understand the problem at hand. Following from this, the specification will be discussed in further detail with the aid of UML diagrams. The solution design and methodology will then be explored before explaining the implementation of the software. Finally, black-box and user testing will be used to evaluate the success of Anxiet-ease, which is followed by a discussion of the project including the options of further expansion.

2.0 Background

2.1 Anxiety

Currently in the UK, anxiety is the most prevalent mental health condition (Kessler et al, 2005). 17% of adults have a diagnosed anxiety disorder according to the NHS survey of Mental Health and Wellbeing in 2016. In primary care mental health settings, as many as 20% of patients have a clinically diagnosable condition (Gask et al, 2018). Anxiety refers to a group of conditions which have similar symptoms and neural responses. Anxiety is a primal response encoded into neural pathways, that evolved as a mechanism to keep us alert to threats in our environment. Anxiety becomes an issue when it occurs in a situation without threat, or when the symptoms are out of proportion for the current level of threat (Gask et al, 2018).

Examples of conditions categorised as anxiety include: Generalised Anxiety Disorder (GAD), phobias, panic disorder, OCD and more. Symptoms of such disorders do vary, however, the most common symptoms across the group of disorders include: hyperventilation, racing thoughts, increased heart rate, and many more (NHS, 2019).

Individuals who suffer from anxiety often have what is known as a trigger. This is a stimulus or situation which can set off the anxiety response in the person's brain, or triggers the anxiety (Hull, 2021). Although there are some common triggers, they are not the same for every person. They can be both overt or discrete, and are not always known before hand (NHS, 2018). Examples of triggers include: loud noises, flashing lights, and crowded places.

While a diagnosis leads to a pathway of intervention, it is not only those with an official diagnosis who experience anxiety. Most people will experience feelings of anxiety during their life. As previously stated, anxiety is a neural response to perceived threat, which is present in all humans. Therefore, most people can benefit from some sort of anxiety relief, at some point in their lives.

2.2 Current Clinical Intervention

The typical pathway for clinical intervention for anxiety in the UK starts with the patient's General Practitioner (GP) . Two options of intervention are offered: Medication or a referral to the relevant mental health service.

Prescriptions will differ dependant on the severity and intensity of the symptoms. When prescribed medication for long term use, patients are offered either a Selective Serotonin Re-uptake Inhibitor (SSRI), or a Serotonin and Noradrenaline Re-uptake Inhibitor (SNRI), both of which work by increasing the amount of serotonin in the brain (Artigas, Nutt and Shelton, 2002). These are more commonly known as antidepressants, but have been found to also be incredibly useful in the treatment of anxiety. However, for the majority of people, this is not the preferred route when experiencing psychological discomfort (Mohlman, 2012). This leaves the option of a therapeutic referral.

As it currently stands, when a patient is referred for psychological support with anxiety, they receive a treatment plan of Cognitive Behavioural Therapy (CBT). CBT is a short term therapeutic course which aims to change the way participants think and respond to their issues (Mansell, 2008). During the program, the therapist teaches the patient strategies and tools to help cope with their anxiety, such as breathing and mindfulness techniques (Hofmann, 2020).

Weekly homework is set during the sessions and involve activities such as journaling, anxiety confrontation, and mood and trigger tracking (Bakker, 2008). The objective of the CBT course is for the patient to be less sensitive to anxious triggers and to also learn how to better interpret, deal, and respond to anxious emotions (Mansell,2008).

There is the limited options of non-clinical intervention through the use of mobile mental health applications. This will be discussed further in Section 2.7.

2.3 Limitations of Clinical Intervention

While both clinical approaches are still greeted with apprehension by the general public, the medicinal pathways hold particular resistance and limitations (Hickie, 2007). There still exists a heavy stigma around medication, especially compared to therapeutic intervention (Mohlman, 2012). This is an obvious source of discouragement for patients in need of

intervention. While highly effective, SSRIs and SNRIs both take a period of time to begin working, and come with a large amount of side effects (such as nausea, dizziness, insomnia and more.). They are also not an instantaneous fix (NHS, 2018., Dunlop and Davis, 2008). However, perhaps the most overlooked limitation with medication is that it is not suitable for patients with mild symptoms, leaving the most effective interventions unavailable to the majority of the population.

The frequently discussed and obvious issue of the underfunded nature of health care providers across the UK, is a clear limitation of current therapeutic intervention. The first notable issue this leads to is long wait lists (Omylinska-Thurston, 2019). Upon referral to their GP, patients are often met by 3 month long wait times (BPS, 2013), and are still not seen by an experienced professional for months following the initial screening to receive intervention (Mind, 2010). Because of the nature of long wait times, GPs and patients are both well documented to be apprehensive about referrals to such services, making only the most severe cases being the ones to get referrals, and patients reluctant to engage with the service. This also brings about another issue of clients having no options for immediate intervention (Collins & Corna, 2018).

The second issue refers to the lack of further support for the patient after completing a short course of CBT, unless they require a step-up in intervention intensity (Williams & Martinez, 2008). This once again leaves patients whose symptoms are not considered severe enough, without an option of intervention. As a consequence of the referral process being less than optimal, patients are either not considered of enough importance for intervention, or the ordeal of attempting to receive intervention leads to the client not wanting to engage with therapeutic practices (Omylinska-Thurston et al., 2019).

2.4 Mindfulness

The goal of a mindfulness activity is to bring attention to a specific feeling or stimulus, allowing a participant to reflect inwards (Kabat-Zinn, 1994). However, the process of engaging in mindfulness intervention can be quite effortful and challenging without guidance (Creswell, 2017). The benefits of routinely engaging in mindfulness activities has been shown to reduce stress, and mental and physical fatigue, while increasing overall positive affect and quality of life (Nyklíček & Kuijpers, 2008). The usefulness of

mindfulness activities as a psychological intervention for anxiety have also been well documented (Hofmann et al., 2010).

Breathing exercises are an example of mindfulness activities. The participant is directed to only focus on their breathing, keeping it in time with a predetermined pattern. For people with anxiety, these exercises are essential, as hyperventilation and breathing issues are extremely common symptoms of anxiety disorders (Huey and West, 1983).

Focusing on an external stimulus of sounds and noises are also considered to be powerful mindfulness techniques and a kind of meditation, helpful in reducing anxiety (Loo, 2018). Nature sounds have especially been shown to help induce a true meditative state. These sounds have been proven to help calm individuals, giving them a sense of peace (Yu, Hu, Funk, and Feijs, 2016). It has also been found that with indoor meditation, a mediative state is achieved quicker in an environment where nature sounds are playing (Hussien Ahmed et al., 2017).

2.5 Flow

Flow is the psychological concept of pure focus, or “being in the zone”. When engaged in flow a person is in the optimal state of energised concentration on the task at hand (Csikszentmihályi, 1975). Flow is becoming so engrossed in an activity, you begin to lose awareness of your external environment, time, and own personal needs. Flow has been described by researchers as the antithesis of anxiety (Csikszentmihalyi, 1990). A loss of reflective self-consciousness, and a sense of personal control or agency over the situation or activity are characteristics of flow. Research into the concept has found that people can experience “feelings of serenity”, worries about the self dissipate, without the interference of the ego causing anxiety (Mao et al, 2020).

In order to achieve flow, an activity needs to posses three key characteristics according to the originator of the theory, Csikszentmihályi (1975). Firstly, there must be a clear goal, this gives structure to the task. Secondly, there must be clear and immediate feedback. This is to allow the person to adjust their performance to maintain the flow state. Lastly, there must be a balance between the perceived difficulty of the task and the persons own ability. This difficulty to skill level ratio is absolutely key in maintaining a flow state; there

has to be confidence in their own ability (Csikszentmihályi, Abuhamdeh,& Nakamura, 2005).

If there is no clear goal to the task, the person can be easily distracted and therefore flow is lost. If there is no feedback, or ambiguity in its delivery, a person can drift out of the flow state as they will not know if they are close to, or have reached the goal. If the task at hand is too difficult for the person's own perceived skill level, it can produce feelings of uneasiness, leading to anxiety (Nakamura and Csikszentmihalyi, 2014). Flow can be difficult to get right, but when done correctly can be harnessed in a variety of ways.

While the theory of flow originated from positive psychology, much of the research into flow has been contained within the occupational and workplace psychological schools of thought. More recently, there has been a huge uptake of the number of studies happening in clinical and medical psychology, with studies showing that even pain perception can be modified using flow (Zavarize, 2016). More recently there has been some discussion into the possibility of harnessing flow to reduce anxiety. Several researchers investigating the relationship between anxiety and flow have found that the flow state is negatively correlated with anxiety (Kirchner et al., 2008). While flow is at its highest, anxiety is at its lowest (Fullagar, Knight & Sovern, 2013).

Research has also indicated that inducing flow appears to be an effective method of reducing anxiety compared to the dominating practises of cognitive behavioural therapy and drug interventions (Fullagar et al., 2013). While a person is engaged in flow, the complete lack of awareness of their surroundings stands to make flow a perfect distraction from anxiety. When a person experiences an anxious episode, being distracted from the triggers in their environment and their own anxious thoughts can become essential. This has also been supported by other researchers who also advocate for the use of flow promoting technology in anxiety prevention and reduction (Mao et al, 2020).

2.6 Distractions and games

Games and distractions are two things that go hand in hand. Video games have been notorious since their inception for flow state. This immersive experience has also been found to be an important aspect of video games as described by gamers and observers (Seah & Cairns, 2008).

As previously discussed, flow has the potential to be the ultimate distraction from anxiety. With this comes the issue of producing flow in a wide range of people, in any environment. Games are the simplest answer (Jin, 2011). Most video games have all three characteristics needed to induce a flow state: clear goals, immediate feedback, and a skill level to difficulty ratio. This example of flow has been seen time and time again when looking back at the popularity of mobile applications over the past decade; Flappy bird, fruit ninja, angry birds, and many more are examples of games that induced a flow state in users. This project proposes harnessing these existing trends and use them for psychological good.

It has been seen in detail throughout multiple research studies that both flow and games provide a distraction adequate enough to distract from anxiety in a variety of settings and situations (Allani & Setty, 2016., Zavarize, et al., 2016). The concentration required to succeed in game play is key to games being a distraction. Therefore, from studies that found gaming to be an adequate distraction from external stimuli, it can be inferred that this distraction was achieved due to the game inducing flow in the player.

For a game to become the ideal distraction for a wide variety of people, there are two characteristics that stand out as the most important: The goal, and methods to achieve the goal, must be clear and simple, and the game must be suitable for wide variety of people with different skillsets. These two characteristics overlap with the three main principles of flow, as well as with each other. Usability, in terms of both application navigation and level difficulties, no matter the users skill set, is essential.

2.7 Existing Implementation

There are a variety of currently available applications that claim to help mental health issues. These applications include: Headspace, Calm, iBreathe and many more (Heaspace, 2021., Calm, 2021., Jade Lizard Software, 2017). Each application usually has its own speciality, targeting a single issue, with a single intervention. Applications such as MindShift or Anxiety Solution (Anxiety Canada Association, 2013., Anxiety PSYT Ltd, 2020) offer practises such as mood tracking and breathing exercises. Headspace is an example of an application which teaches guided mediation. When searching online for applications that help with anxiety, articles will also include recommendations for a variety

of simple gaming applications for their distraction properties. While not psychologically designed, general populations have found these applications useful when dealing with episodes of anxiety (Cabot and Wilkinson, 2016).

While there have been many documented strengths about such applications, there still remains skepticism surrounding their legitimacy, which, in some cases, is well founded. Research has found the legitimacy of these applications to be varied (Sucala et al,. 2017). Researchers concluded that there is potential for successful intervention through the use of mobile applications. However, it also highlighted the need for the applications to be researched based, with many applications advertising their effectiveness without any research in the techniques they employed (Marshall, Dunstan, and Bartik, 2019).

Applications need to start being developed from a research based background instead of app stores being saturated with applications with outlandish and unfounded benefits. This is the first clear limitation. There needs to be a sense of legitimacy within the field, allowing users to trust the application to be useful. This is imperative in getting an application aimed at reducing anxiety, to work for a wide variety of people.

Most applications that have been researched based are not free to download. This creates the obvious monetary barrier to evidenced based intervention by excluding the majority of the potential user base. Most of these monetised applications also include an extensive sign-up process, and involve data collection. Not only does data collection discourage the user from engaging with and trusting the application, a registration and sign in process creates a barrier between the user and accessing the tools they need in the height of their anxious episodes.

There is also an issue of repetition in this field, with a lot of the current applications only having one function that has been done many times before. This limits the user base to those whose symptoms fit within a subset of symptoms. If the current coping strategies that exist are not helpful or are limited, the person is left without an option for an application intervention.

Another option for people who suffer from anxiety is mobile games. However, these games are not developed through a psychological lens or specifically for people with mental health conditions. This can lead to the presence of overwhelming stimuli, causing

sensory overload, triggering anxiety (Brennan, 2021). Adult colouring is another form of distraction available, but it lacks the ability to induce flow in a wide population as it is hard to create a skill to difficulty level ratio in such an activity.

There still remains a clear three fold gap in the market of mental health applications. Firstly, there is yet to be an application which harnesses the potential of flow as a distraction for anxiety. Forcing users to go to less evidence based means if this is their desired intervention. Secondly, there remains a distinct lack of applications which offer users a range of activities, in different styles of intervention. Leaving a huge user base ignored. Lastly, the range of evidence based intervention is mostly monetised, pricing many users out of intervention.

2.8 Mobile Application Development

In 2020 alone, 218 billion mobile applications were downloaded. This follows an ever growing trend which is increasing each year (Perez, 2021). Mobile application development is the process of designing and producing such applications. While following the main principles and practises of software development, mobile development can be seen as its own distinct discipline. The applications developed for mobile phones are designed to work on mobile specific operating systems such as Android and iOS.

Many different operating systems have tried and failed to achieve market dominance, but none have come close to Android and iOS since their initial releases. In the UK market, operating system demand is split by .01 of a percent between iOS and Android. However, globally Android holds 72% of the market share, with Apple's iOS holding only 27% (Stat counter , 2020).

2.8.1 Android Development

Initially released in 2007, Android is now the most popular mobile operating system. Android is based on a modified version of the linux kennel. The kernel works as an application programming interface which provides a method of communication between the software of an application and the hardware of the device it is being run on (Bovet and Cesati, 2005).

When it comes to software used to develop Android applications, programmers have a wide variety of IDE's to choose from. There is often discussion on which is the best choice, but little can be decided in terms of a general conclusion still. It comes down to personal preference. However, in general discussion the IDEs most frequently mentioned include Android Studio and Eclipse.

Eclipse was initially released in 2001 (eclipse tools project, 2001). While being written in Java, and being designed primarily for Java users, it is an incredibly customisable environment supporting many languages including: C, C++, JavaScript, Python and many more (Watson and DeBardeleben, 2006). Before the release of Android Studio, Eclipse was the official IDE for the operating system (Wolfson and Felker, 2013). The downsides of Eclipse come from its usability, with the lack of a layout editor. The fact it is not designed specifically for Android development is also seen as a negative when being discussed in this particular context.

In 2013 came Android studio, replacing Eclipse as the official IDE for Android (Hohensee, 2013). It was built using IntelliJ, but has been designed specifically for Android development (Ducrohet, Norbye and Chou, 2013). It supports Java, Kotlin, C++ and XML. Android studio uses a gradle to compress the source files of the application and produce an APK file (Android package) to run it. Android studio has it own virtual device called the emulator, which allows users to run their application on any Android device from their desktop. It has a layout editor as well as pre-built templates. Overall, it makes the programming process easier for developers. The drawbacks of this software include the relatively high RAM and CPU usage, and the limit to only Android development (Yener and Dundar, 2016).

2.8.2 Android languages

The language choice for Android application development lies between two main options: Java or Kotlin. Both of which have, at some point been, the official language for the operating system (Lardinois, 2019). While being released almost 20 years apart, the differences in the languages are slight.

Initially released in 1995, Java is a high-level object oriented programming language. (Poo, Kiong and Ashok, 2007). It was developed on a write once run anywhere ethos, to allow for cross platform implementation (Gosling and McGilton, 1996). The Java code is

compiled into byte-code and run through the Java Virtual Machine (JVM), which acts as an interpreter between the system hardware and the code itself. Java is one of the most popular coding languages to date maintaining its dominance over the decades since its inception (Dai, 2019).

In 2011 JetBrains announced the development of Kotlin, which was released in 2016. It is a cross platform general purpose programming language intended to work alongside Java, using the JVM also (Mateus and Martinez, 2019) . Kotlin was designed to create a more concise syntax, whilst still being capable of all the functionalities Java has to offer.

The two programming languages are about as similar as you can get in terms of capabilities, the difference coming down to 3 main components. Firstly, the length of code. Kotlin code is usually shorter (Ardito et al., 2020). Secondly, the run time. Java has been found to be faster than Kotlin (Glukhov and Mullayanov, 2020). And lastly, the documentation. The popularity of Java as a language leads to a more accessible and higher volume of documentation (Bose, 2018).

2.8.3 Android Strengths and Weaknesses

One of the biggest strengths of Android comes from its product diversity. The Android operating system is run on phones from a variety of different companies, giving more choice for the consumer from features to price range (Android, 2021). The diversity of Android manufactures leads inherently to innovation. With so many different teams working on so many different products, Android has been at the forefront of mobile innovation.

The open-source nature is another important strength of Android to note, especially in comparison to its main competitor's (iOS) closed source nature (Sheikh et al., 2013). Open source provides transparency and user customisation, creating a community of diverse developers who are able to freely contribute towards the design of the system (AlMarzouq et al., 2005).

Security is the first weakness of note to discuss when exploring Android (Dhama, 2014). The nature of the open source system can allow for hackers to find and exploit the systems vulnerabilities. Although the company has done much to try and quell these concerns, since before the official release these concerns have been well vocalised in the

developing community. In 2018, Android announced the gaps in security in comparison to iOS were closed, ensuring Android phones are just as hard to hack as their main competitors, Apple (Hautula, 2018).

Perhaps one of the biggest draw backs of the Android operating system is the lack of update support. Unlike its main competitor, Android cannot install software updates directly to devices due to their multiple manufactures. This has implications for not only the previously mentioned security issues, but also user satisfaction, with update schedules varying per manufacturer (Google, 2021).

2.8.4 Mobile Architecture

There are three main kinds of mobile app architecture: native, hybrid and mobile web apps (Tun, 2014). The success of any mobile application greatly depends on the selected architecture. The performance needed from the application is the determining factor when selecting the architecture for an application, with each option holding individual strengths and weaknesses.

A native application is stored and run on the device. They are built specifically for the operating system they are intended to run on (Charland and Leroux, 2011). For example, for an Android application, Java and Android studio are needed to create the native software. If the application was built using Java, and required porting to a device running on iOS, the developer would need to recreate the app using Swift and AppCode (JetBrains, 2011). Native applications work best for apps with rich visuals, and offline apps among others (Lim, 2015). Advantages of native apps include the maturity and documentation of the method, but its real strength comes from the performance of the User Interface (UI) visuals and the overall speed (Serrano, Hernantes and Gallardo, 2013). Native applications also have access to the in-built capabilities of the device. The biggest negative of native apps is the need to re-write the applications code for each platform. This makes deployment slower, and a developer with a larger skillset would be needed (Malavolta, 2016).

A hybrid application combines native and mobile web applications, having access to the hardware of the device while being based on the web. This allows for applications to be run across platforms, meaning a developer would not have to rewrite the code for both Android and iOS (Que, Guo and Zhu 2016). The advantages of this architecture centres

on the “write once, run anywhere” philosophy. The applications developed do not need to be rewritten for each platform (Malavolta et al, 2015). The negatives of hybrid applications concern the simplicity of the UI capabilities as well as the varying performance capabilities across devices (Biørn-Hansen, 2020). Additionally, while the code might only need to be written once, it will need to be debugged across all platforms with less tools to do so (Khandeparkar, Gupta and Sindhya, 2015).

Mobile web applications are accessible through URL in a browser, however, an icon can be assigned to it on the home screen. The application is not stored on the device, but is stored on the web and ran in a browser (Serrano, Hernantes and Gallardo, 2013). Mobile web apps work best for basic functionality and UI applications, that do not need to store a lot of data offline, and mobile versions of websites. The strengths of this method are that the code does not need to be rewritten for each platform, and they are available from engines directly (Holzer and Ondrus, 2012). However, the negatives concern the limited performance capabilities overall (Sin, Lawson, and Kannoorpatti, 2012).

3.0 Project Specification

3.1 Analysis and specification

The goal of this project is to create an evidence based mobile application that helps users ease the feelings of anxiety. This will be achieved through offering users a variety of activities designed through a psychological lens, utilising theories and research such as flow and guided meditation. The project will have been a success if the application proves fit for purpose. This will be decided through user testing. If the user analysis proves that Anxiet-ease has been effective in lowering their overall “level” of anxiety, before and after using the application, and users believe it has had an overall psychologically positive effect, the application is fit for purpose.

As discussed in Section 2.7, there is currently clear issues in the market of psychological applications. Most centrally the issues lie with a lack of empirical evidence through the designing process, as well as a lack of choice for the user base. Anxiet-ease will overcome these issues through the design of a selection of different activities, based on current theory and interventions. This will allow for the user to guide their own evidenced based intervention, choosing what works best for them.

The activities can be divided into two categories: mindfulness and distractions. This project proposes to harness flow as a method of reducing anxiety through the use of games. This will distract the users from their anxious thoughts and triggers. The mindfulness side of the application will utilise more traditional therapeutic methods, using the research evidence from practices such as guided meditation and breathing, and sound therapy. Additional tools and softwares will be needed to implement these methods, such as Adobe After Effects for animation purposes and Audacity for sound manipulation.

Anxiet-ease is a native Android application. The choice of a native application is due to the advantages it holds, especially its user interface. With gaming distractions being a huge part of this application, it was important to ensure a smooth graphical interface with little lag and better capabilities. While actual deployment would be made more difficult with this method, Android was chosen to ensure a wider target audience could be reached. Future expansion of the project would tackle cross platform implementation.

As for software, Android Studio was the choice to develop this project. With the application being developed solely for an Android operating system, Android Studio was the obvious choice being the official IDE of the OS. The emulator holds a special allure here, being able to test Anxiet-ease on Android devices.

Anxiet-ease will be developed in Java. Choosing Java as the language comes down to two main reasons: run times and documentation. The superior run times of the language compared to Kotlin along side the accessibility of documentation outweighed the endorsements of competitors. The aesthetics of the application will be designed using XML. This was chosen for the simplicity in its syntax and its ease of collaboration with the rest of the project choices. One of the strengths of using Android studio is the XML drag and drop features of layout objects, however, for the purposes of this project this feature was not utilised in order to further develop my skills as a programmer.

3.2 Use Case diagram

This use case diagram shows the functionalities of the system in relation to the user. These functional requirements along with the non-functional requirements are detailed in Appendix A.

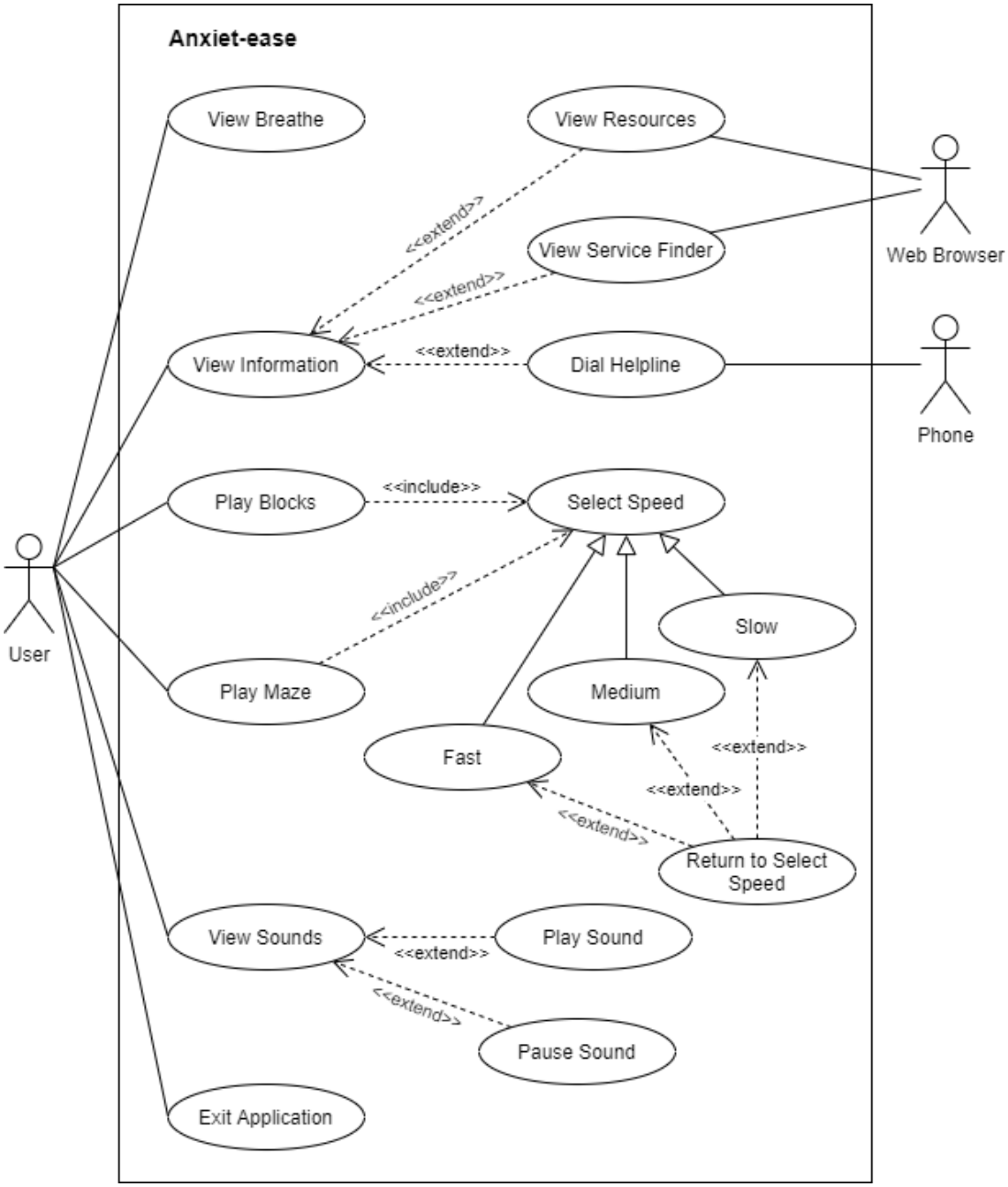


Figure 1: Use Case Diagram

4.0 Solution design

4.1 Overall application

First and foremost, the entire application must have an easy to use, intuitive design. This is to ensure any populations of anxious individuals will not be excluded from the benefits of the application. There will be a variety of distraction and mindfulness activities. This will allow users to guide their own interventions, making choices that work best for them. The colour scheme of the application will follow the principles of colour psychology as well as user experience (UX) design. 3 colours were chosen: green, purple and blue. Each colour was chosen for the calming properties they have be shown to elicit (Logan-Clarke and Appleby, 2009)

4.2 System Architecture

The architecture of the system is a component of software engineering that must not be overlooked. It is a model that helps define the structure of the system, providing a formal description and representation of the system. There are many different styles of software architecture. The following segment outlines the two most preferred for mobile development, MVC and MVP, as well as their suitability for different softwares.

MVC stands for Model - View - Controller. This pattern was designed to make the development of large pieces of software easier. The Model contains the data, The View is what is displayed to the user and the Controller is the way in which the model and the View are connected, updating the model when there is a change in the view and vice versa. In this architecture, the model also has direct access to the view (Deacon, 2009). When it comes to Android development, this method becomes difficult, as it is easy to blur the lines between the controller and view components.

The MVP model stands for Model - View - Presenter. This alternative approach overcomes the issues associated with the MVC method (Gu and Tang, 2010). This structure provides an easy way for developers to envision the structure of their applications, providing modularity and testability. This method also keeps the code cleaner, and a lot easier to maintain. In this model, instead of having a singular controller class, the MVP model allows for a one-to-one mapping between presenters and views.

The responsibilities of the controller within the MVC model is split between the presenter and the view in the MVP architecture.

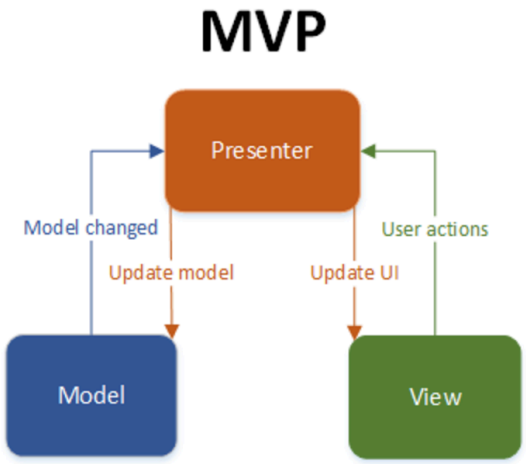


Figure 2: MVP architecture

The chosen software architecture for this project is the MVP model. This particular model was chosen for the modularity it provides. The presenters are mapped one-to-one to a view, which allows for the modularity and scalability of this particular methodology. The MVP model will also allow for an easier implementation of new features through the use of interfaces. Given the compactness of the current application, MVP architecture is by far the best fit. All of the view classes inherit from the view interface. This allows them to present the visual information to the user. The presenter classes inherit from the presenter interface. The presenters handle all of the logic of the application (e.g. the onClickListeners for the buttons) for each of the views. The model carries the application resources and ID's, which the presenters use to reference resources. The inheritance of these methods via the interfaces contributes to the modularity and scaleability of the application. The performance of the application is hindered through the added layer of abstraction, but a balance between performance and scalability was reached, without affecting the user's experience of the system.

The class diagram below (Figure 3) illustrates the MVP architecture of the system along with the connections between each of the classes. Each view extends from the IView interface and is controlled through a presenter, which extends from an IPresenter interface. The views have no connection to the resources, but are populated via the presenter calling upon the ID's of each of the items in the resource folder.

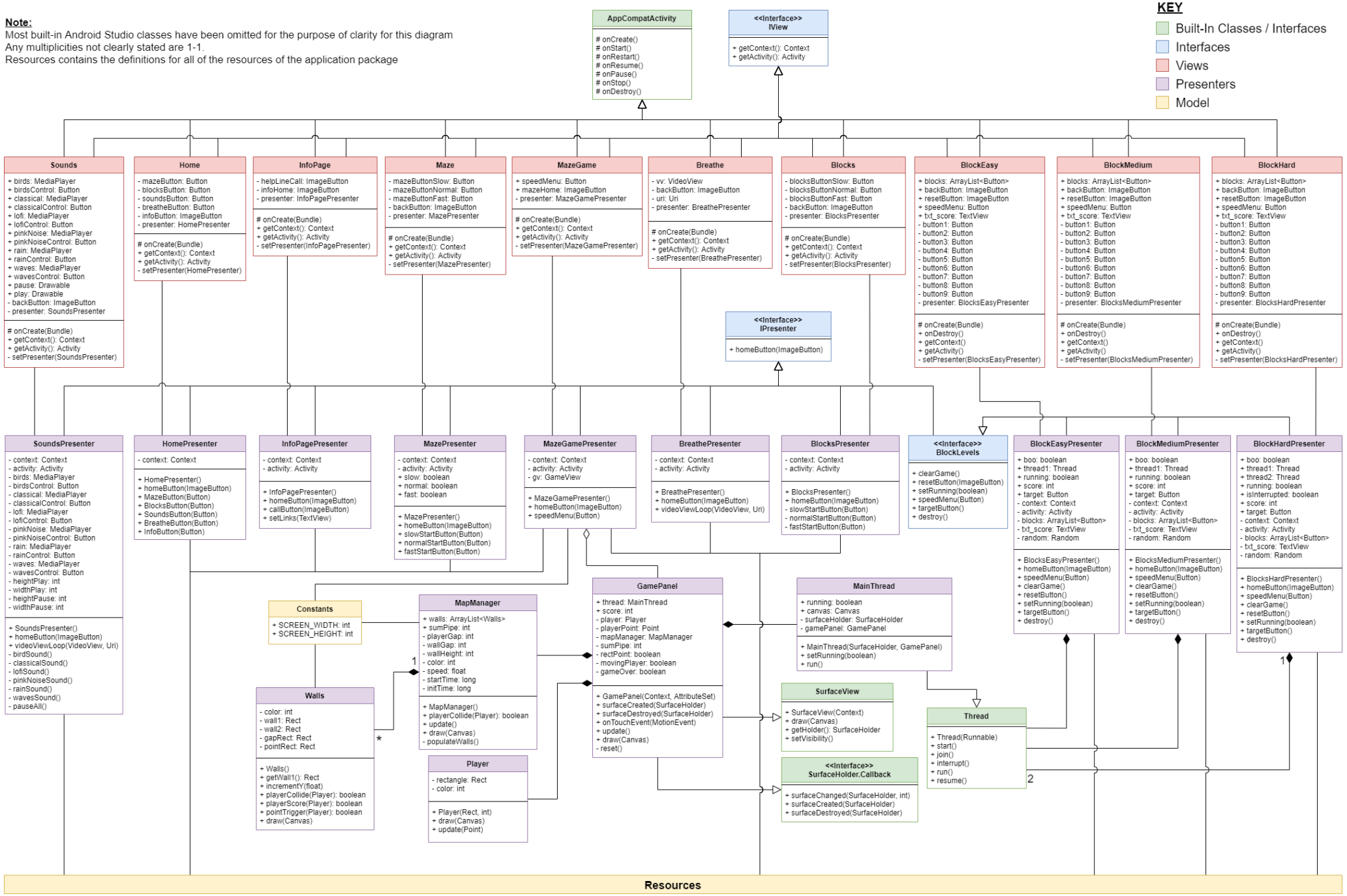


Figure 3: UML Class Diagram

4.3 Home and Info

The application will open to a simple and easy to use home page, with buttons to direct the user to each of the activities they may choose from. There will also be a further information section. This follows good psychological practice, directing users to mental health services and charities, also providing a helpline for those users who may be in crisis. It is important psychologically to let users know Anxiet-ease is not the cure, it is a tool to help. Therefore, it is essential to give users the tools to access clinical intervention when needed.

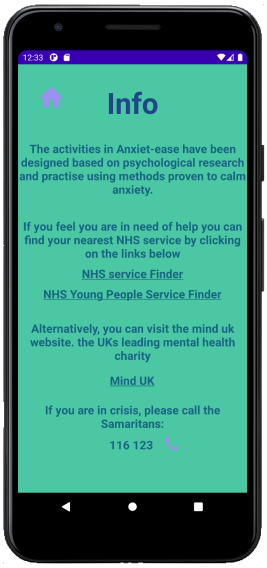


Figure 4: Information activity

4.4 Breathe Activity

The first mindfulness activity chosen for the application was the breathe activity. The intent behind this comes directly from cognitive behavioural therapy breathing practises. One of the most common symptoms of anxiety is hyperventilation and breathlessness (Abraham, 2020). The breathe activity eases user's anxiety through the use of guided breathing.

When the user starts the activity, they will be shown instructions and an animation. The animation is of a shape slowly expanding, and collapsing. This animation will continuously loop until the user exits the activity. The instructions tell the user to follow the animation to guide them through some deep breaths. Users will inhale slowly with the expansion of the shape, and exhale slowly as the shape collapses.

Figure 5: Breathe animation

The animation will have a time of 10 seconds for the total loop. This will help time the deep breathing of users, helping ease them out of a state of hyperventilation. Even without the overt psychological symptoms of hyperventilation, the practice of daily deep mind-full breathing holds positive psychological effects for all populations. (Cho et al., 2016., Feldman, Greeson and Senville, 2010)

4.5 Sounds Activity

The design of the sounds activity is based in the mindfulness principles of focusing on a specific stimulus. With this in mind the sounds activity is a collection of sounds with evidence illustrating a multitude of beneficial effects. When the user selects the sounds activity on the home screen they will be taken to a collection of sounds, clearly titled with a play/pause button. The sound will continuously loop once played, this will not stop until the user pauses the sound, plays a different sound, or exits the activity or application. A varied collection of sounds was essential here to appeal to a wide variety of users. The chosen sounds can be split into two categories: natural and man-made sounds.

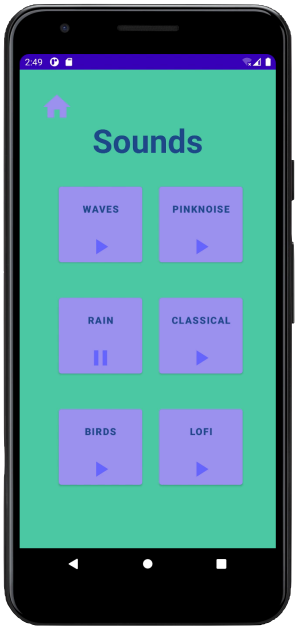


Figure 6: Sounds activity

Nature based sounds have a vast array of existing research, which highlight the benefits of nature sounds, not only for mindfulness (e.g. guided meditation), but also for daily human life (Ray et al., 2020). Of the chosen nature sounds, the importance of variety must once again be stressed. Therefore, the final sounds chosen are: Waves, birds, and rain. These are all sounds repetitively studied, and well known for their calming effects (Cutshall et al., 2011, Chiang, 2012).

The man-made sounds in the activity are perhaps much better at demonstrating this variety. The first chosen sound here is pink noise. It is a relatively newly studied sound, with its better known relative, white noise, being more extensively researched and its divisive nature well documented. Pink-noise is a few decibels lower than white noise, and a better census can be reached on its calming effects (Lai et al., 2021)

Lo-Fi music is another relatively newer concept that has been utilised in this activity. The creation of this genre was based around audio to improve and facilitate concentration. While there remains very little in terms of peer reviewed research, general consensus of listeners points towards the lowering of anxiety and improvement in concentration (Johnson, 2020).

The last choice of sound for the activity was classical music. This genre has had a long evidenced based history of being a calming sound. The range of research into this genre include investigations of concentration, intelligence and personality. Research includes the well documented effects of violence and aggression reduction associated with classical music (Krahé, and Bieneck, 2012). The qualities of the genre, such as slow soft melodies, with simple repetitive beats and harmonies, make the musical product a great tool for relaxation and calming.

It Is also important to highlight the physiological effects of sounds and music when designing the implementation of this activity. Music has long been proven to have physiological effects on listeners such as influencing the heart rate (Esche and Evequoz, 1999). A song with a high beats per minute (BPM) will raise a persons heart rate, and a low BPM will decrease it. With this in mind during the selection of the sounds, any sound over 100BPM was disqualified from consideration, as increased heart rate is a symptom which contributes to the overall feelings of anxiety. Instead, both of the musical sounds will range between 60-80BPM, which is the average heart rate zone (NHS, 2021). This would give the effect of lowering the heart rate of the user, easing one of the most common symptoms of anxiety.

4.6 Maze Activity

The intention behind the maze game is concerned with the previously discussed concept of utilising flow as a distraction for anxiety. As such, it is imperative the maze contains three qualities: a clear goal, immediate feedback, and a skill to difficulty ratio.

When playing the maze game, the user must guide a square player through a gap in walls that are scrolling down the screen. Once the player successfully clears the gap, a point will be awarded at the top of the screen. Walls will continue to generate until the user either collides with the walls, or presses the home or speed menu buttons. If the player collides with any of the walls on screen, the game will stop and instruct the user to tap anywhere to try again. If the user taps the screen, the game will reset to its initial state, and start again.

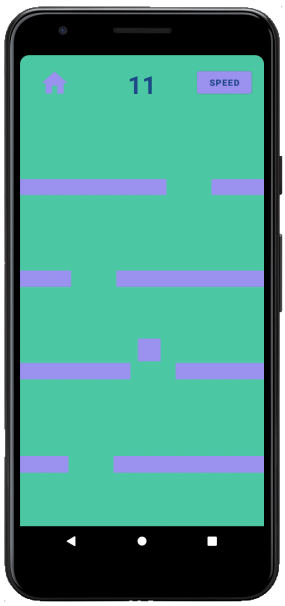


Figure 7: Maze activity

The goal of getting through the gaps in the walls is explained to the user before starting the game. There is no other functionalities of the game, making the instructions and the goal extremely clear and easy to communicate.

The user receives immediate feedback from the game. The constant moving of the obsticles are a type of visual feedback, the game over screen provides instructional feedback, and the score is a type of positive feedback. This gives the user several ways in which they can assess their performance to maintain flow.

When the user clicks on the maze game, they are first taken to the speed menu containing instructions to the game and three speed options. The user can change the speed of the walls scrolling down the screen which provides the skill level to difficulty ratio.

4.7 Blocks Activity

The blocks activity has a simple premise. To get a point, the user must click target block from a set of 9 blocks. The target block is made clear to the user by turning yellow, while the other blocks remain purple. As soon as the user taps the target block, it will award a point, and change back to purple.

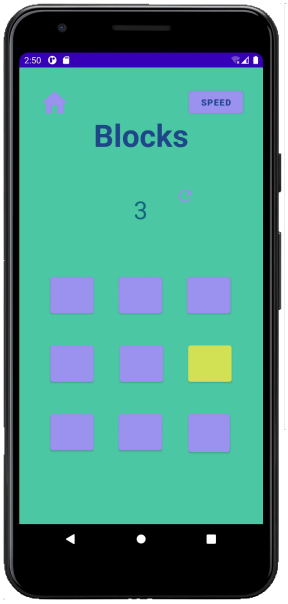


Figure 8: Blocks activity

The blocks game maintains flow using the core flow principles in a similar manner to the maze game. There is a clear goal of clicking the target block, there is immediate feedback through the points and visual state, and a skill to difficulty ratio is maintained through difficulty levels.

The levels to the game could not be implemented using a tradition style of increasing the speed as the difficulty increases. It would not have made sense for the style of game play. Instead, a more creative solution was found to incorporate difficulty levels, including separate psychological intentions. The slow speed is intended to slow users down by including a delay between the deactivation of the target, and the activation of the next target button. When the user taps the target there is a delay in the light up of the next

target. They cannot continuously tap, and must keep their concentration of the screen to wait for the next target. The target will not move until the user interacts with it.

The medium mode differs from slow mode as there is no delay in generating a new target block. This is to allow users to set their own pace, incase they lose their concentration. After the user taps the target block, it will deactivate, and immediately activate the next target. The target will not move until the user taps it.

The hard mode is to ensure there is a challenging option, for those that need high difficulty to achieve flow. The level follows a “wac-a-mole” style protocol. The target block is activated for a brief period of time, in which the user can interact with it and gain a point. If the user does not select the target in time, the target is deactivated, before presenting a new target after a short delay. This level requires high concentration and fast reflexes, keeping more skilled users engaged with the activity.

5.0 Project Management and Methodology

Throughout the entirely of the project following good software engineering, and psychological practice is essential. On the software side, using principles such as DRY (don't repeat yourself) and KISS (keep it simple, silly) ensuring concise and easily readable code. Testing of the code on both the software and user ends were also of particular importance in following good software engineering practice. In terms of good psychological practise, extra care and consideration was taken during the creation and development of the application. One such instance to illustrate this, is the information page of the application, outlining the limitations of the application, and providing users with further resources to access clinical intervention. Further to following endorsed philosophies and principles, a software development methodology was essential to the planning and execution of this project.

The agile software development method follows an iterative process, developing increments of the software in cycles (Edeki, 2015). This process allows for developers to release software iteratively, allowing them to release software benefits and improvements quicker and more frequently. Agile development is a methodology followed by a wide number of companies, due to its ability to keep up with continuously changing software requirements.

The waterfall method follows a more linear and rigid flow. Each phase of development has a distinct focus and goal (Balaji and Murugaiyan, 2012). Each goal must be 100% complete before moving on to the next phase of development. The linear nature of this model allows for the method to be easily understood and managed, so can be beneficial to even the most inexperienced of teams. This method works best when the project has clear and unchanging requirements for the software. The waterfall method does not have a way of going back through the project to change the direction or modify the project. This method has an extremely rigid structure, which can be both slow and costly.

DevOps is a method which is not just focused on the development methodology, but also contains a set of practises designed to help support organisational culture. This process encourages a change of procedures to allow for a better collaboration between the departments responsible for the different sections of development, e.g. development, UX, etc. The strengths of the DevOps method surround the speed of the development

process. Products are quick to market, and thanks to the cross team collaboration, bugs and issues are resolved quickly, leading to higher customer satisfaction. However, the speed of this process can sometimes be obstructed due to the need of human interactions for some phases of project development and testing.

Taking into consideration the above mentioned methods strengths and weaknesses, the agile development method was chosen. The iterative methodology allowing for a modular process, with the capability of changing each activity as the functionality and requirements change and grow throughout the process, was of particular strength in the consideration process. The rigidity of the waterfall method was not compatible with the typical processes of game development. The required functionalities of the application rendered this methodology incompatible with the project. While the speed of the DevOps method is a big strength, the nature of the project being psychologically based, and the potential implication of the user base, as well as psychological ethical considerations, leaves this method also incompatible with the development of this project.

Forward thinking and future planning is essential for embarking on an extensive research and development project such as this. A project timeline is a key tool needed to achieve this. With the nature of the project having potential issues with the game development as well as user testing, the adoption of the agile method allows for the creation of a flexible timeline. This allows for the development process to loop back and between implementation goals, to achieve a flawless final product. The timeline of this project is split between three main tasks: research, development, and the report.

The research holds particular importance given the basis of the application, ensuring sound psychological theory and implementation. Therefore, a substantial amount of time and effort must be given here in the initial planning of the project. The development is the process which is the most time consuming and continuous aspect, where the iterative nature of the agile methodology will be of particular use. The importance of allowing enough time to write the report itself cannot be overlooked during this process, thus the report will be developed throughout the research and development, and testing processes. A Gantt project schedule chart is included in Appendix B.

6.0 Solution Implementation

6.1 Overall Application

The UI of the application was developed using XML, which is a markup language used to design the layout of each view in Android studio, similar to HTML for web design. The view in the MVP architecture consists of Java code used to create the visuals on screen using this XML layout. The layout may consist of objects, such as buttons, which can be accessed by the classes via their ID's, to attach functionality to the object (i.e. adding a listener to a button). The ID's of the layouts and objects are stored within the resource folder and are called by the presenter classes to populate the view and provide functionality.

An activity begins by calling the startActivity() method, where an Intent consisting of the current activity and new activity is passed through (Android, 2014). This new activity is added to the top of the activity stack, which is shown to the user, and its life-cycle is initiated. The onCreate() method is called first, where the content of the view is set through the ID of the XML layout of a particular view. The activity can be paused and resumed via onPause() and onResume() methods. Upon completion of an activity (e.g. wanting to return to the home page), the finish() method is implemented. When this method is called, the activity's life-cycle ends, wherein the onDestroy() method is invoked to destroy the instance of the view. As the life-cycle of one activity ends, a new life-cycle begins of a different activity. A state diagram showing transitions between activities can be seen in Appendix C.

6.2 Home and Information

The implementation of the home and information pages were the first challenges of the project. The home page has 4 buttons, and 1 image button (Appendix D). Once the user clicks on one of the 5 buttons, the home page starts a new activity, taking the user to one of the following activities: breathe, sounds, maze, blocks or info.

By clicking on the image button indicating further information, the user is navigated to the information activity. Here they are presented with 4 options of interaction with the activity: click the service finder links, click the further resources link, call the help line, or return to the home page. When clicking on either of the service finder links or the further resources

link the process is the same, starting with the manipulation of the string resource folder. In the string resource file, a href link was assigned to the strings shown to the user. The setMovement method can then be called within the InfoPagePresenter Java class, to automatically take the user to the devices web browser, loading the relevant link.

The helpline number is shown on the screen to the user via a text view. Next to the number is an image button of a drawable image asset of a phone. When the user clicks the phone button, a new ACTION.DIAL Intent is created. startActivity() is called and the created intent is passed through it. This opens the phone application on the users device, with the number for the helpline already dialled, ready to call.

For the user to return to the home page from the information page, they must click on the home button. This is also an image button with a drawable assigned to it, only this time the image asset is of a home icon. When this button is clicked, finish() is called on the activity to take the user home. This completely shuts down the activity, as the last method called in the life cycle.

6.3 Breathe Activity

When the user clicks on the breathe button on the home screen, they are taken to a new activity holding a home button, a text view of instructions, and an animation for the user to breathe along with. The initial creation of this section of the application used a gif, created using the software Gimp. While the implementation itself was sound, the quality in terms of the animation was not sufficient for the overall appearance of the application. Upon expansion and collapse of the circle, the shape edge would become pixelated due to the reduced frame rate. The implementation was incredibly simple here, needing only a gif in the XML file, and the only Java being the back button's finish() command.

```
public void homeButton(ImageButton button) {
      button.setOnClickListener(v -> {
            activity.finish();
      });
}
```

Listing 1: Home button implementation

To further improve the aesthetics of the activity to fit with the rest of the application, an mp4 animation was instead created using the timeline animation software Adobe After Effects (Adobe, 1993). The loop of a circle expanding and collapsing was created through

using key frames to animate to the different states. The key states were: fully expanded circle with 100% opacity, and a fully contracted circle at 0% opacity. The contraction and expansion of the circle was achieved by animating the scale property from 1 to 0. Several versions of the animation were exported in different file formats such as .gif, .mp4, and .mov to see which format had the best performance, file size and colour handling. The various advantages and drawbacks of each format were explored here. For example, while .gif supports background transparency, this comes at the cost of file size and smoothness of animation. An mp4 file does not support transparency, but optimised the animation and file size, ensuring faster loading and a smaller footprint in the project.

The Breathe and BreathePresenter classes populate and control the view presented to the user. The onCreate() method finds the video view and also assigns the mp4 file from the raw resource folder to the view. Then the video is set to start, and loop. There are no user controls associated with the video view, so the user can focus only on their breathing. The user can return to the home page to select other actives via the home button.

```
vv = (VideoView)findViewById(R.id.breathevid);
backButton = findViewById(R.id.breathehome);
uri = Uri.parse(“Android.resource://
com.example.anxiet_ease/" + R.raw.anxbreathe);
presenter.homeButton(backButton);
presenter.videoViewLoop(vv, uri);
```

Listing 2: Breathe video implementation

6.4 Sounds Activity

One of the many packages available to import in Android studio is the MediaPlayer. The sounds activity makes use of several media players to provide the user with the calming sounds. There are 6 sounds in total, each assigned to a media player, which in turn are controlled through their own buttons. Each of the buttons for the sounds make use of image assets to communicate to the user the state of each sound; playing or paused.

There was also the option of using a sound pool instead of media players. As all of the sounds were relatively short and were intended to loop, a sound pool would have been a suitable option for the implementation of this activity. However, as sound pools are designed for sound effects, and the choice of multiple play buttons for each sound, it was decided that media players were the better option in this instance.

The evidence based sounds were all edited through the Audacity software. This was done for two reasons. Firstly, each of the sounds needed to be of a similar volume. As they were taken from different creators and are very different categories of sounds, the original volume of each of the sounds varied. If the user clicked the waves sounds, for example, and then clicked the pink noise sound, the increase in volume would have been very jarring. This is obviously not pleasant for user experience in general, but can also be considered an anxious trigger. Therefore, the amplify tool was used in Audacity to increase or decrease the volume of each track. The second reason for editing the sounds in Audacity was for the purposes of looping. It was difficult to find sounds clips within the chosen categories that looped seamlessly. Audacity was used here to edit the sound loops for a more discrete loop. If the user were to listen to the sound for a long while, there will be little disruption, or indication of elapsed time.

For the Java code of the activity, after finding the XML of the Class, the onCreate() method for this activity creates the media players, assigning the correct sound for each of them. Next, the buttons for each sound are found, along with the image assets for play and pause. These have been saved as a drawable resource in the application. The play image is assigned to each button, as none of the sounds will yet be playing. All of the sounds are controlled through their own buttons, each have a listener which is assigned through the SoundsPresenter class.

Each of the methods for the sounds follow a similar format. The setOnClickListener() method is used on each of the buttons, with the result being controlled by an if statement. Using the media player's isPlaying() method, the system can understand the user's intent when pressing the button. If the media player is playing, pause() is called. The drawable on the button is also assigned to play, as the sound is no longer playing.

```
wavesControl.setOnClickListener(v -> {
      if (waves.isPlaying()) {
            waves.pause();
            wavesControl.setCompoundDrawables( null, null, null, play );
      } else {
            pauseAll();
            waves.setLooping(true);
            waves.start();
            wavesControl.setCompoundDrawables( null, null, null, pause );
      }
});
```

Listing 3: Sound buttons control

In the else of this if statement, the media player will not be playing, therefore the user intends to play the selected sound. The start() method is called on the media player. The media player is also set to continuously loop until user input pauses it once again. An issue arose during the implementation of this method. When a user is playing a sound, and then plays another without pausing the first one, both will play simultaneously. This is unpleasant for user experience, and would require more effort from the user if they were forced to pause the first sound before playing the second.

To overcome this issue simultaneously playing multiple sounds, multiple if statements were used. Each if statement checked if a particular sound was playing before pausing it, if the condition was true. Such a long set of if statements is required here due to the lifecycle of Android media players. You cannot call pause on a media player that has never been played before, so the status of each player must first be checked, instead of using one check and calling pause on all of them. The drawable resource must also be changed back to the play image asset, if any of the other media players are paused as a result of this method call.

```
private void pauseAll() {
      if (waves.isPlaying()) {
            waves.pause();
            wavesControl.setCompoundDrawables( null, null, null, play );
      } if (rain.isPlaying()) {
            rain.pause();
            rainControl.setCompoundDrawables( null, null, null, play );
      } if (birds.isPlaying()) {
            birds.pause();
            birdsControl.setCompoundDrawables( null, null, null, play );
      } if (classical.isPlaying()) {
            classical.pause();
            classicalControl.setCompoundDrawables( null, null, null, play );
      } if (pinkNoise.isPlaying()) {
            pinkNoise.pause();
            pinkNoiseControl.setCompoundDrawables( null, null, null, play );
      } if (lofi.isPlaying()) {
            lofi.pause();
            lofiControl.setCompoundDrawables( null, null, null, play );
      }
}
```

Listing 4: Sounds pauseAll() method

When the user clicks the home button on the sounds activity, the finish() method will be called, taking the user to the home page. But due to the use of the media players in the activity, release() must be called on each media player before finish() is called. This is to

free up memory, and allow the application to run faster and smoothly. Calling release when exiting the activity will also stop any of the sounds which might have been playing.

6.5 Blocks Activity

When the user clicks the blocks game on the home page, a new activity begins, containing the speed options for the game, along with the instructions for the user on how to play. Each of the buttons for the speed options of the blocks game start a new activity, the controlling classes of which correspond with the button options. However, these activities do not have an XML layout file of their own, as is traditional with activities. As the layout for each of the levels is the same, one XML file was created and assigned as the layout for the 3 separate level activities of the blocks game.

In the onCreate() method of each of the three levels, the buttons are found on the XML and they are then added to an array of buttons, called blocks. A new random is also created here, used to select the target button. A random button is chosen from the blocks array each time a new target is required. This button is the target button. The target button will change colour, from purple to yellow to communicate to the user it is the target button. Upon being clicked a point will be added. An activity diagram in Appendix C illustrates the flow of actions.

The initial implementation for the game lit up a random target button and assigned a point once clicked, but in order to loop the game, recursion was needed. This lead to overflowing the stack incredibly quickly due to the nature of the game. Therefore, a complete redesign on the approach taken was needed. Eventually, threads with a controlling while loop was decided on as the optimal approach.

A thread is used to loop the game, the implementation of which differs per level. Once the target button is clicked, a point is added to the users score, and a boolean is set to true. While this boolean is true the target is deactivated, and the thread sleeps for 5 seconds in the easy level. Once the thread sleep is over, a new target button is assigned. The boolean is then set to false, and and the while loop restarts. The implementation for the easy level was almost identical to the implementation of the medium level. In the medium level, the thread does not sleep for 5 seconds like in the easy level. This removes the

delay between the input and the generation of a new target. The new target is assigned immediately after the previous target is clicked by the user.

```
Runnable clickButtons = () -> {
      while (running) {
            target.setOnClickListener(v -> {
            score++;
            txt_score.setText("" + score);
            boo = true;
            });
            while (boo) {
                  target.setBackgroundColor(Color.parseColor("#9c91ed"));
                  target.setOnClickListener(null);
                  try {
                        Thread.sleep(4000);
                  } catch (InterruptedException e) {}
                  target = blocks.get(new Random().nextInt(blocks.size()));
                  target.setBackgroundColor(Color.parseColor("#d4e263"));
                  boo = false;
            }
```

Listing 5: Blocks easy mode loop

The implementation of the hard level differs from the previous two levels as this level utilises two threads instead of one. Following a “wac-a-mole” style of play, the target block must be automatically reassigned if the user does not click the button within a specified amount of time. The first thread of the hard level is the same as the thread in the medium level, but when the target is clicked, instead of just adding a point and setting a boolean to true, it also interrupts the other thread.

```
Runnable clickButtons = () -> {
      while (running) {
            target.setOnClickListener(v -> {
                  thread1.interrupt();
                  score++;
                  txt_score.setText("" + score);
                  boo = true;
            });
```

Listing 6: Blocks hard mode thread interruption

The second thread provides the auto-move capabilities in the hard level. Unless the thread is interrupted through user input, it will sleep for 1 second before setting the controlling boolean to true. This then deactivates the current target, and activates a new one.

```
Runnable timerButtons = () -> {
      while (running) {
            try {
                  if (isInterrupted) {
                        Thread.sleep(2000);
                        isInterrupted = false;
                  }
                  Thread.sleep(1000);
                  boo = true;
                  Thread.sleep(2000);
            } catch (InterruptedException e) {
                  isInterrupted = true;
            }
      }
```

Listing 7: Blocks hard mode auto-move thread

When a user wants to exit the game, they can either go to the home page, or to the speed menu. Taking the user home cannot be accomplished in the same way as the mindfulness activities with just calling finish(). The thread(s) of the activity must first be stopped. This is done by setting the setRunning() method to false. The controlling boolean is set to false, finish() can then be called. startActivity() must be called after calling finish for the blocks game, as finish would take the user back to the previous activity, which in this case is the speed menu and not the home page. Therefore, the methods for the home and speed menu buttons are almost identical, but the home method starts the home activity. The speed menu method can end with finish() as the blocks game was started from the blocks speed menu activity.

6.6 Maze Activity

Just like the blocks game, when a user clicks the maze game on the home screen, they start an activity which contains instructions for the game, and speed level options. Unlike the blocks game, the maze does not need to have three separate classes for the speed levels. This is because the speed operates in a more conventional manner, with the rest of the mechanics of the game staying the same. When generating the walls in the maze, a speed must be passed into the wall object. Therefore the speed levels can be in the same class, controlled by booleans set on the speed menu. When the user selects a speed, the MazeGame activity starts, the player is in the starting position, and the walls start to generate. This will continue until the player collides with one of the walls of the maze, or clicks the home or speed menu options. An activity diagram in Appendix C illustrates the flow of actions in the maze game.

The game itself is drawn to a canvas, this is what allows the movement of the walls, which would not have been possible through the sole use of XML. With this method, there is no way for the inclusion of buttons to navigate to the rest of the application. Therefore, the game is a View within an XML layout. Within the XML layout are the buttons for the user to navigate home, and to the speed menu, as-well as their score. The score is in the XML to ensure that the user can see the score unobscured by the walls of the maze which generate below it.

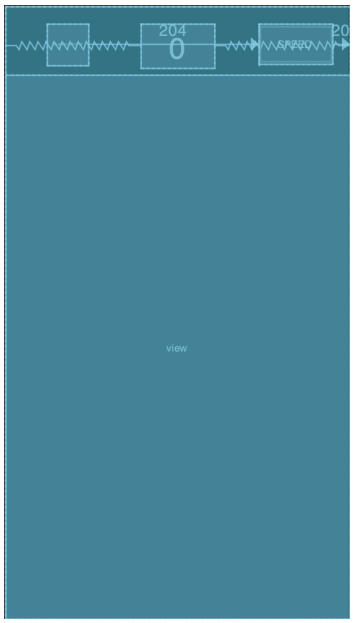


Figure 9: Maze game XML

The walls and the player of the game are drawn to the canvas using a Rect object. The walls of the maze are first generated upon the start of the activity. The walls appear to scroll down the screen by moving incrementally down each frame. The distance is calculated by multiplying the speed, by the amount of time between frames. The player can be moved around the entirety of the maze, this is achieved through the use of a point assigned to the centre of the rect. When the user drags the square player around the screen, the playerPoint is updated with the coordinates of the movement, which therefore moves the square player also. The choice of Rects for both the player and the walls of the maze was down to the intersect() method that can be used on them. This is a boolean that will return true when any of the specified rects intersect each other. In this case, if the player and any of the walls intersect at any point the boolean returns true, and the game over method is triggered. This stops the movement of the game, and displays a message to the user to tap anywhere on the screen to try again. When the user taps on the game over screen, the game reset method is called, reinitialising the state of the maze back to the starting state.

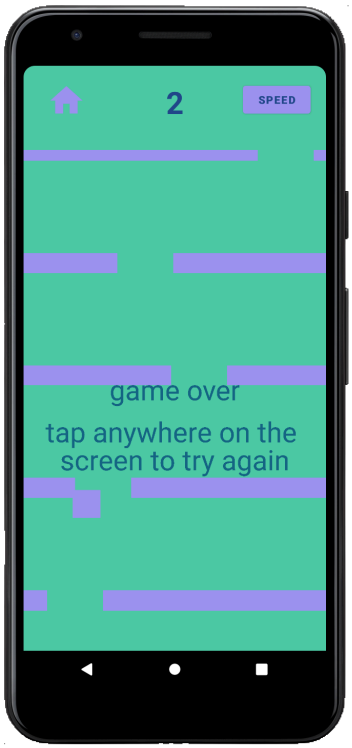


Figure 10: Maze game game over screen

The scoring element of the activity was a problem of high complexity. The free movement of the player, between an arbitrary part of the screen is hard to track and score. The solution that was landed on was through the use of the same intersect() method which is used to trigger the end of the game. When the walls of the maze are generated it appears there is only two rectangles (one for each side, with a gap in between them for the player). This is not the case. The walls are actually generated in groups of 4 rectangles, two of which are hidden. This 4 rect design (Figure 11) is responsible for the scoring system of the game. The third rect is in the gap of the two walls, and the fourth is in the middle of the current walls and the next. When the player intersects with the 3rd rect, a boolean is set to true. When the player intersects with the 4th rect and the boolean is true, 1 point is added and the boolean is set to false. The player must not be able to intersect with both the 3rd and 4th rect simultaneously, or this will continually add points.

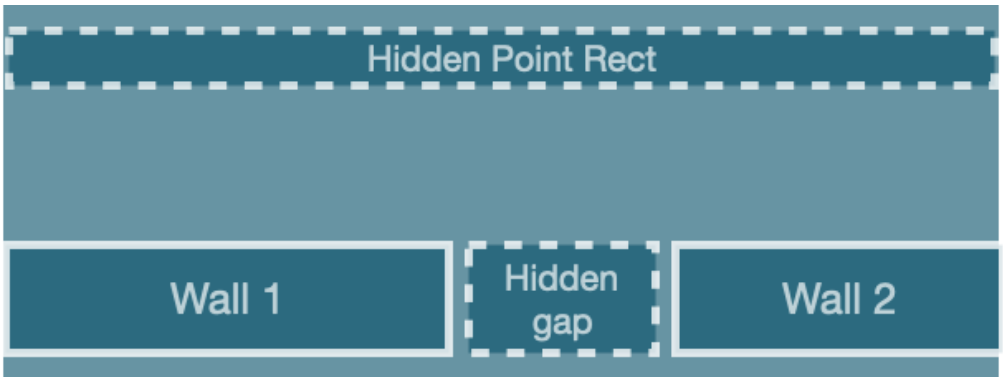


Figure 11: Maze scoring system blueprint

The design process went through many versions of the points system within the game. However, they all had a compromise on the users overall experience. One such instance of this was adding a point when a wall is removed at the bottom of the screen. The reasoning behind the rejection of this method was that the point should be for clearing the obstacle, which is done as soon as the player has completely cleared the gap, not when the wall is removed at the bottom of the screen. Another version of points system involved restricting player movement to side to side only, using a timer to allocate the points. However, this had massive implications for the overall game play, as the free movement of the player throughout the screen is definitely preferable.

Other versions of the maze game had also been designed before the decided upon solution, such as a maze using a recursive backtracking algorithm. The maze would randomly be generated from a grid through the removal of certain cell walls. The player would start in the top left of the maze, and have to navigate to the target cell at the bottom right hand side of the maze. While this would have perhaps been the more conventional “maze” to deploy, the continuous nature of the current implementation was better for achieving the goals of this project and inducing flow.

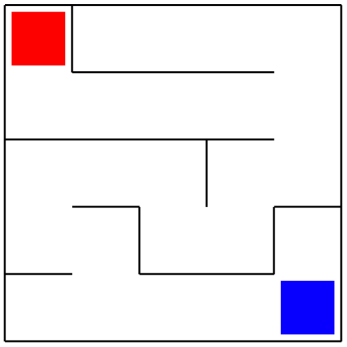


Figure 12: Recursive backtrack algorithm maze

7.0 Testing and Evaluation

7.1 Testing

Testing is one of the most important aspects of software development, ensuring a perfectly running system. Given the allotted time frame of the project, traditional JUnit testing could not have been accomplished with the scale of the research and implementation needed. Therefore, each element of the application has been manually tested through the use of black box and user testing.

Black box testing is a software testing process focusing on the validation of the requirements of the software (Nidhra and Dondeti, 2012). Black Box is a relatively quick method of testing that does not require any knowledge of the code or the implementation of the application. The only knowledge needed is the functionalities of the application and the expected outputs. Test cases are formulated for each functionality of the application, determining the inputs and expected outputs of user actions. The functionality passes the test when the expected outputs are produced.

Black box testing was completed before the user evaluation took place, to ensure a high quality user experience. There was also an opportunity for users to report any bugs that may have been missed during the testing process. No bugs or performance issues were discovered. However, it should be noted that performance on the Android studio emulator differs from the performance on an actual Android device. Details of the black box testing process, including the test cases, can be found in Appendix E.

7.2 User Evaluation

To assess the success of this project, a user evaluation was imperative. Ideally user evaluations would have been completed with the use of a focus group or interview to enable a more open-ended qualitative analysis. However, due to the constrains of the current pandemic, a user questionnaire was instead developed.

A case study of 10 people with diagnosed and undiagnosed anxiety, with Android mobile devices, took part in a user research study. Anxiet-ease was imported on to each participants phone for them to use at their own discretion. After a few days of the participants using the application, they were sent an online questionnaire to find out the user's perceptions of the application, and the overall usefulness and psychological legitimacy.

The questionnaire consisted of 16 questions (Appendix F), two of which requiring a qualitative response. More than half of the participants strongly agreed that Anxiet-ease had a positive effect on their anxiety. The rest of the users all agreed, with only one participant remaining neutral about the effect of the application. The response is even more overwhelmingly positive with the usability of the application, with the majority of participants strongly agreeing that Anxiet-ease is easy to use, and the rest agreeing.

The strongest activity in the survey was the maze game, with 100% of participants strongly agreeing that it provided a good distraction for anxiety. The majority of participants also said that it was their favourite activity within the application. The rest of the respondents chose the blocks game as their favourite. This clearly shows the user's preference for the use of flow as a tool to reduce anxiety.

In terms of the conclusion to make from this analysis, firstly, the application proves fit for purpose. Users reported an average 3 point reduction in anxiety (on a 10 point scale) before and after using Anxiet-ease. User's rated the distractions as the favoured activities within the application to help ease their anxiety. This provides more evidence to the growing idea of utilising flow as a means of anxiety intervention.

Secondly, through the qualitative analysis of the user's free responses, the wide variety of choice offered by Anxiet-ease as opposed to traditional psychological applications is not only a correctly identified gap in the current market, but a gap that Anxiet-ease is extremely successful in filling. The other previously identified gap in the market can also be inferred to have been filled, through the response to the blocks and maze games. Users have overwhelmingly agreed on the usefulness and favouritism of these distraction activities, citing their ability to keep them distracted. This gives evidence of the flow state being induced in a wide variety of people.

Lastly, the responses from the questionnaire indicate a strong positive reception not only to the application, but the thought of further expansions. Many of the respondents indicated that they would enjoy if the application were to have an even bigger range of activities, which could be implemented in future iterations of the project. Further research would need to be conducted due to the sample size of the population to determine the true level of usefulness of the application as a whole. This would also aid in understanding what elements the users would like the application to develop further when expanding the application.

8.0 Discussion

This project provides further proof to the growing evidence base of the usefulness in harnessing flow in psychological intervention. The overwhelmingly positive results found in the user feedback analysis show the potential of flow as an intervention for anxiety. Further research is needed to confirm the effects can be extrapolated to a wider audience, and variety of other mental health conditions.

Further research implications include the potential for researched based mental health applications. A convenience focused and medically under funded society leads to a demand for virtual intervention. This project joins the growing evidence base advocating for psychologically designed mental health applications. More time, research and resources need to be dedicated to exploring this intervention option, which is proving to be of enormous benefit to afflicted populations.

The success of Anxiet-ease also leads to potential implications for CBT practise. The expansion of the application into a clinical therapeutic aid seems to be an obvious step. Using the proven successes of Anxiet-ease, an application can developed to work alongside CBT, optimising the care of patients. By digitising CBT tools (e.g. ABC journals), patients and clinicians can better track progress. This could also provide users with similar care, both during and after intervention, while simultaneously improving intervention.

The strengths of the project include the ease of use, fitness of purpose, and the polished nature of the software engineering and psychological methodologies. These factors lead to the development of an end product that is not only psychologically designed, but professionally polished and ready for market. Anxiet-ease achieves every goal laid out in the initial stages of development, showing the project to be an overwhelming success.

The biggest drawback during the development process of this project was the time constraints. Ideally, the application would have contained a wider variety and choice of activities. The limited time frame for this project made this impossible.

Upon reflection of the final product and user feedback, the design of the mindfulness activities would be changed if given the opportunity again. The novelty of the activities

within the application is clearly one of the strongest appeals, this novelty does not apply as much to the mindfulness activities which have been implemented before in other applications. Therefore, more original ideas would have been preferred here, though the benefits of the breathe and sounds activities are not be be dismissed. I would combine these into a singular activity, adding another to take its place, such as worry time (Figure 14).

In terms of the language chosen for the development process, the status of Kotlin as the official Android development language may cause the use of Java to be considered a drawback of this project. However, while the code produced might have been more concise, the improved run times provided by Java show this to be superior choice for the purposes of this project.

8.1 Future Expansion

Further expansion for this project holds a plethora of possibility. The afore mentioned research implications hold limitless opportunities to develop not only Anxiet-ease, but a variety of other mental health applications, targeting different issues. This would bridge the ever growing gap in access to intervention.

In terms of the expansion of the current application, cross platform implementation is the first area of expansion needed for this project. The choice of a native application was the correct choice to make when accounting for performance and goals of the application. However, due to the time frame of the project, native implementations across platforms would not have been possible. The choice of Android development was due to market dominance, and accessibility. With proof of concept, and fitness of purpose established, an implementation of the application in Swift for iOS devices would be the next step.

After achieving cross platform implementation, the next expansion to the application would include developing further activities. Adding more activities will keep the existing user base engaged with the application, upholding its therapeutic benefits, while also appealing to and being able to induce flow in a wider population. Below are two examples of activities which could expand the application. Figure 13 shows the Stacks distraction activity. Users must move the stack side to side to catch the falling blocks. Figure 14 shows the activity Worry Time, this activity is a digitised version of the CBT worry time practise (Delgado et al., 2010).

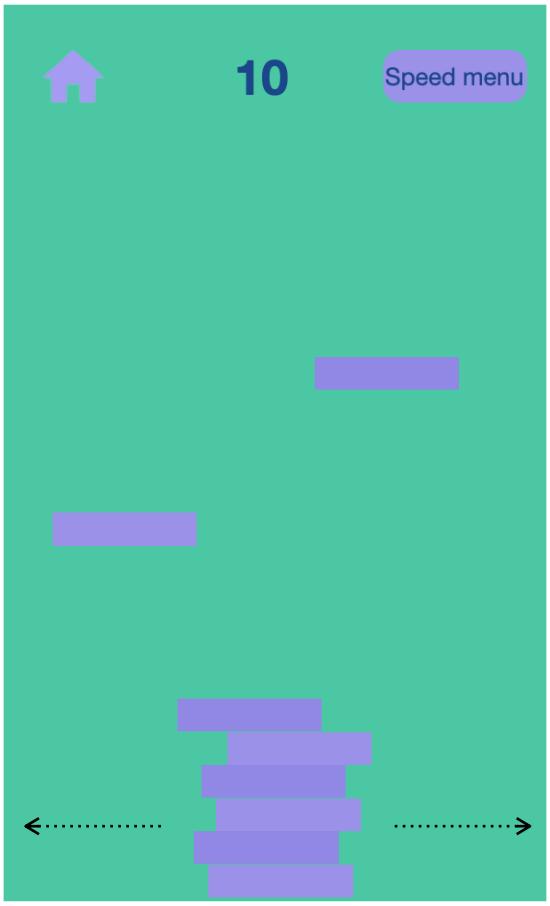


Figure 13: Example of a distraction game to further expand the application, “Stacks”

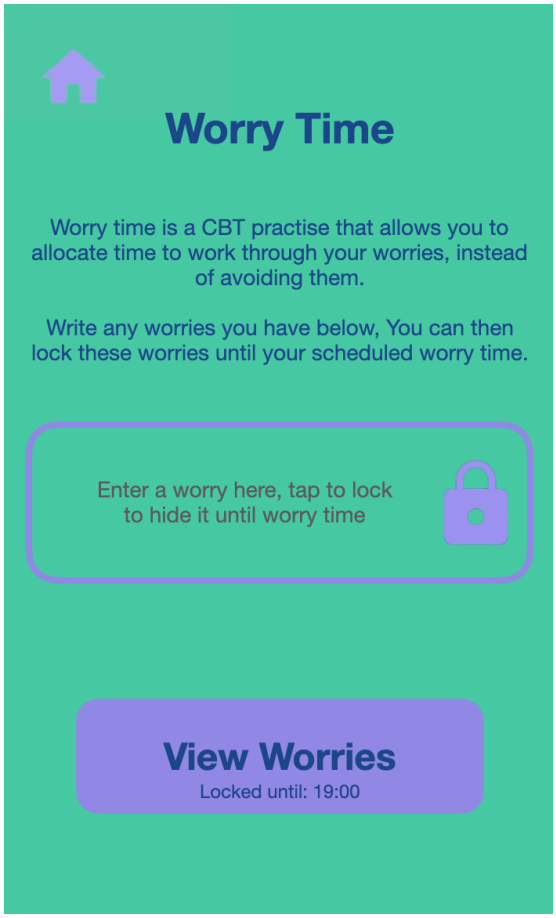


Figure 14: Worry time, a digitised version a a CBT technique

9.0 Conclusion

The main aim of this project was to create a novel mobile application to help ease the symptoms of anxiety - Anxiet-ease. The application created proved fit for purpose, through user evaluation and system testing, implementing mindfulness and distraction activities as a means for mental health intervention. This project has proven to be a success from both a psychological and software engineering standpoint, achieving every aim and goal set fourth.

The user evaluation clearly outlines the benefits of the application, with an overwhelmingly positive response in terms of overall positive psychological effects. The application of flow as a distraction from anxiety is a particular success of note, due to the early stages of existing research in this area of application. A novel approach to intervention has proven to be highly rewarding in terms of the success of this application, as well as the possibilities this holds beyond this project.

From a software engineering perspective, the smooth running and high performance of the application, as well as the cleanness of the code, are particular successes to note. The addition of the distraction games proved to be a huge challenge for the development process that was successfully overcome. The user experience evaluation also found that not a single user experienced an issue with the running of the application.

The clear multiple avenues for expansion of the application hold a lot of possibility for both the computer science and psychological communities. This project shows a clear need for further collaboration between the two disciplines, combining empirical research and tried and tested interventions with the technological creativity of software engineering.

References

Abraham, M. 2020. *Hyperventilation: *The* Anxiety Attack Symptom.* [online] Available at: <https://www.calmclinic.com/anxiety/symptoms/hyperventilation> [Accessed 16 September 2021]

Adobe. (1993). AfterEffects.

Allani, S. and Setty, J.V., 2016. Effectiveness of distraction techniques in the management of anxious children in the dental operatory. *IOSR J Dent Med Sci*, *15*(10), p.69-73.

AlMarzouq, M., Zheng, L., Rong, G. and Grover, V., 2005. Open source: Concepts, benefits, and challenges. *Communications of the Association for Information Systems*, *16*(1), p.37.

Android. 2014. *Activity.* [online] Available at: <https://developer.android.com/reference/android/app/Activity> [Accessed 22 September 2021]

Android. 2021.*Phones and Tablets.* [online] Available at: < https://www.Android.com/intl/en_uk/phones-tablets/> [Accessed 16 September 2021]

Anxiety Canada Association. (2013). MindShift.

Ardito, L., Coppola, R., Malnati, G. and Torchiano, M., 2020. Effectiveness of Kotlin vs. Java in Android app development tasks. *Information and Software Technology*, *127*, p.106374.

Artigas, F., Nutt, D.J. and Shelton, R., 2002. Mechanism of action of antidepressants. *Psychopharmacology bulletin*, *36*, p.123-132.

Bakker, G., 2008. *Practical CBT: Using functional analysis and standardised homework in everyday therapy*. Australian Academic Press.

Balaji, S. and Murugaiyan, M.S., 2012. Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, *2*(1), p.26-30.

Biørn-Hansen, A., Rieger, C., Grønli, T.M., Majchrzak, T.A. and Ghinea, G., 2020. An empirical investigation of performance overhead in cross-platform mobile development frameworks. *Empirical Software Engineering*, *25*, p.2997-3040.

Bose, S., Mukherjee, M., Kundu, A. and Banerjee, M., 2018. A comparative study: java vs kotlin programming in Android application development. *International Journal of Advanced Research in Computer Science*, *9*(3), p.41.

Bovet, D.P. and Cesati, M., 2005. *Understanding the Linux Kernel: from I/O ports to process management*. " O'Reilly Media, Inc.".

Brennan, D. 2021. *What Is Sensory Overload With Anxiety?* [online] Available at: < https://www.webmd.com/balance/what-is-sensory-overload-with-anxiety> [Accessed 15 September 2021]

Cabot, S. and Wilkinson, B., 2016, September. Using mobile-based games as a means for the self-treatment of depression and anxiety in youth. In *Joint International Conference on Serious Games,* p.128-133. Springer, Cham.

Calm.com, inc. 2021. Calm.

Charland, A. and Leroux, B., 2011. Mobile application development: web vs. native. *Communications of the ACM*, *54*(5), p.49-53.

Chiang, L.C., 2012. *The effects of music and nature sounds on cancer pain and anxiety in hospice cancer patients* (Doctoral dissertation, Case Western Reserve University).

Cho, H., Ryu, S., Noh, J. and Lee, J., 2016. The effectiveness of daily mindful breathing practices on test anxiety of students. *PloS one*, *11*(10).

Collins, N. and Corna, L., 2018. General practitioner referral of older patients to Improving Access to Psychological Therapies (IAPT): an exploratory qualitative study. *BJPsych bulletin*, *42*(3), p.115-118.

Creswell, J.D., 2017. Mindfulness interventions. *Annual review of psychology*, *68*, p.491-516.

Csikszentmihalyi, M., 1990. *Flow: The psychology of optimal experience* (Vol. 1990). New York: Harper & Row.

Csikszentmihalyi, M., Abuhamdeh, S. and Nakamura, J., 2005. Handbook of competence and motivation. *New York: Guilford Press, Chapter" Flow*, p.598-698.

Csikszentmihalyi, M. and Csikszentmihalyi, I., 1975. Beyond boredom and anxiety San Francisco. *CA, US: Jossey-Bass.*

Cutshall, S.M., Anderson, P.G., Prinsen, S.K., Wentworth, L.J., Olney, T.L., Messner, P.K., Brekke, K.M., Li, Z., Sundt III, T.M., Kelly, R.F. and Bauer, B.A., 2011. Effect of the combination of music and nature sounds on pain and anxiety in cardiac surgical patients: a randomized study. *Alternative Therapies in Health & Medicine*, *17*(4).

Dai, R., 2019. Java Basics. In *Learn Java with Math* p.19-22. Apress, Berkeley, CA.

Deacon, J., 2009. Model-view-controller (mvc) architecture. *Online][Citado em: 10 de março de 2006.] http:// www.jdl.co.uk/briefings/MVC.pdf.*

Delgado, L.C., Guerra, P., Perakakis, P., Vera, M.N., del Paso, G.R. and Vila, J., 2010. Treating chronic worry: Psychological and physiological effects of a training programme based on mindfulness. *Behaviour research and therapy*, *48*(9), p.873-882.

Dhama, S., 2014. An overview of security challenges of Android apps permissions. *International Journal of Information and Computation Technology. ISSN 0974-2239*, *4*(4), p.373-380.

Ducrohet, X., Norbye, T. and Chou, K. 2013. *Android Studio: An IDE built for Android.* [online] Available at: < https://Android-developers.googleblog.com/2013/05/Android-studio-ide-built-for-Android.html> [Accessed 16 September 2021]

Dunlop, B.W. and Davis, P.G., 2008. Combination treatment with benzodiazepines and SSRIs for comorbid anxiety and depression: a review. *Prim Care Companion J Clin Psychiatry*, *10*(3), p.222-228.

Eclipse Tools Project. 2021. [online] Available at : <https://archive.eclipse.org/eclipse/downloads/drops/R-1.0-200111070001/> [Accessed 15 September 2021]

Edeki, C., 2015. Agile software development methodology. *European Journal of Mathematics and Computer Science*, *2*(1).

Elgin, B. 2005. *Google Buys Android For Its Mobile Arsenal* [online] Available at: <https://web.archive.org/web/20150228085314/http://www.bloomberg.com/bw/stories/2005-08-16/google-buys-Android-for-its-mobile-arsenal> [Accessed 15 September 2021]

Escher, J. and Evequoz, D., 1999. Music and heart rate variability. Study of the effect of music on heart rate variability in healthy adolescents. *Praxis*, *88*(21), p.951-952.

Feldman, G., Greeson, J. and Senville, J., 2010. Differential effects of mindful breathing, progressive muscle relaxation, and loving-kindness meditation on decentering and negative reactions to repetitive thoughts. *Behaviour research and therapy*, *48*(10), p.1002-1011.

Fullagar, C.J., Knight, P.A. and Sovern, H.S., 2013. Challenge/skill balance, flow, and performance anxiety. *Applied Psychology*, *62*(2), p.236-259.

Garg, S. and Baliyan, N., 2021. Comparative analysis of Android and iOS from security viewpoint. *Computer Science Review*, *40*.

Gask, L., Kendrick, T., Peveler, R. and Chew-Graham, C.A. eds., 2018. *Primary care mental health*. Cambridge University Press.

Glukhov, D.V. and Mullayanov, B.I., 2020, October. The Performance Evaluating of Kotlin and Java Implementations. In *2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)* p.1-7. IEEE.

Google. 2021.*Check and update your Android version.* [online] Available at: <https://support.google.com/Android/answer/7680439?hl=en-GB> [Accessed 17 September 2021]

Gosling, J. and McGilton, H. 1996. *The Java Language Environment.* [online] Available at: <oracle.com/java/technologies/language-environment.html> [Accessed 17 September 2021]

Gu, M.X. and Tang, K., 2010, July. Comparative analysis of WebForms MVC and MVP architecture. In *2010 The 2nd Conference on Environmental Science and Information Application Technology* (Vol. 2), p.391-394. IEEE.

Hautula, L. 2018. *Google: Android is just as secure as the other guys.* [online] Available at: <https://www.cnet.com/tech/mobile/google-Android-is-just-as-secure-as-ios-iphone-david-kleidermacher/> [Accessed 16 September 2021]

Headspace Inc. (2021) Headspace.

Hickie, I.B., Luscombe, G.M., Davenport, T.A., Burns, J.M. and Highet, N.J., 2007. Perspectives of young people on depression: awareness, experiences, attitudes and treatment preferences. *Early intervention in psychiatry*, *1*(4), p.333-339.

Hofmann, S.G., 2020. *The Anxiety Skills Workbook: Simple CBT and Mindfulness Strategies for Overcoming Anxiety, Fear, and Worry*. New Harbinger Publications.

Hofmann, S.G., Sawyer, A.T., Witt, A.A. and Oh, D., 2010. The effect of mindfulness-based therapy on anxiety and depression: A meta-analytic review. *Journal of consulting and clinical psychology*, *78*(2), p.169.

Hohensee, B., 2013. *Getting Started with Android Studio.*

Holzer, A. and Ondrus, J., 2012. Mobile app development: Native or Web?. In *Proc. Workshop eBus.(WeB)*

Huey, S.R. and West, S.G., 1983. Hyperventilation: its relation to symptom experience and to anxiety. *Journal of abnormal psychology*, *92*(4), p.422.

Hull, M. 2021. *Anxiety Triggers.* [online] Available at: <https://www.therecoveryvillage.com/mental-health/anxiety/related/anxiety-triggers/> [Accessed 10 September 2021]

Hussien Ahmed, M.M., Silpasuwanchai, C., Salehzadeh Niksirat, K. and Ren, X., 2017, May. Understanding the role of human senses in interactive meditation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems,* p.4960-4965

Ionescu, D. 2012. *Original Android Prototype Revealed During Google, Oracle Trial.* [online] Available at: <https://www.pcworld.com/article/254539/original_Android_prototype_revealed_during_google_oracle_trial.html> [Accessed 16 September 2021]

Jade Lizard Software. 2017. iBreathe.

JetBrains. (2011). AppCode.

Jin, S.A.A., 2011. “I feel present. Therefore, I experience flow:” A structural equation modeling approach to flow and presence in video games. *Journal of Broadcasting & Electronic Media*, *55*(1), p.114-136.

Johnson, S. 2020. *How lo-fi artists make music perfect for work. (Or studying. Or chilling.).* [online] Available at: <https://www.washingtonpost.com/lifestyle/wellness/how-lo-fi-artists-make-music-perfect-for-work-or-studying-or-chilling/2020/02/18/c88bf92e-49e4-11ea-bdbf-1dfb23249293_story.html> [Accessed 16 September 2021]

Kabat-Zinn, J., 1994. *Wherever you go, there you are*. Hachette Books.

Kessler, R.C., Berglund, P., Demler, O., Jin, R., Merikangas, K.R. and Walters, E.E., 2005. Lifetime prevalence and age-of-onset distributions of DSM-IV disorders in the National Comorbidity Survey Replication. *Archives of general psychiatry*, *62*(6), p.593-602.

Khandeparkar, A., Gupta, R. and Sindhya, B., 2015. An introduction to hybrid platform mobile application development. *International Journal of Computer Applications*, *118*(15).

Kirchner, J.M., Bloom, A.J. and Skutnick-Henley, P., 2008. The relationship between performance anxiety and flow. *Medical Problems of Performing Artists*, *23*(2), p.59-65.

Krahé, B. and Bieneck, S., 2012. The Effect of Music–Induced Mood on Aggressive Affect, Cognition, and Behavior. *Journal of Applied Social Pschology, 42(2),* p. 271-290.

Lai, H., Wang, G., Gao, M., Zhou, Y., Wu, S. and Wang, J., 2021. A New Product Development and Effect Analysis of Tinnitus Therapy Based on Pink Noise Tone.

Lardinois, F. 2019. *Kotlin is now Google’s preferred language for Android app development.* [online] Available at: <https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-Android-app-development/ > [Accessed September 16 2021]

Lim, S.H., 2015. Experimental comparison of hybrid and native applications for mobile systems. *International Journal of Multimedia and Ubiquitous Engineering*, *10*(3), p.1-12.

Logan-Clarke, V. and Appleby, J., 2009. What is color therapy. In *Color Therapy Healing Workshops. Bognor Regis, UK.*

Loo, M., 2018. *Combining nature sounds or music in mindfulness activities: Their effects on mindfulness attentional skills, wellbeing and session attendance rates* (Doctoral dissertation, Murdoch University)

Malavolta, I., 2016, October. Beyond native apps: web technologies to the rescue!(keynote). In *Proceedings of the 1st International Workshop on Mobile Development,* p.1-2.

Malavolta, I., Ruberto, S., Soru, T. and Terragni, V., 2015, June. End users' perception of hybrid mobile apps in the google play store. In *2015 IEEE International Conference on Mobile Services,* p. 25-32. IEEE.

Mallik, A. and Russo, F., 2021. The Effect of Music & Auditory Beat Stimulation on Anxiety.

Mansell, W., 2008. What is CBT really and how can we enhance the impact of effective psychotherapies such as CBT?. *Against and For CBT: Towards and Constructive Dialogue*, p.19-32.

Mao, Y., Yang, R., Bonaiuto, M., Ma, J. and Harmat, L., 2020. Can Flow Alleviate Anxiety? The Roles of Academic Self-Efficacy and Self-Esteem in Building Psychological Sustainability and Resilience. *Sustainability*, *12*(7), p.2987.

Marshall, J.M., Dunstan, D.A. and Bartik, W., 2019. The digital psychiatrist: in search of evidence-based apps for anxiety and depression. *Frontiers in Psychiatry.*

Mateus, B.G. and Martinez, M., 2019. An empirical study on quality of Android applications written in Kotlin language. *Empirical Software Engineering*, *24*(6), p.3356-3393.

McDirmid, S. and Odersky, M., 2006. The scala plugin for eclipse. In *Proceedings of Workshop on Eclipse Technology eXchange (ETX)*.

Mind. 2010. *We need to talk: getting the right therapy at the right time* [online] Available at: <https://www.mentalhealth.org.uk/sites/default/files/we-need-to-talk.pdf> [Accessed 10 September 2021]

Mind. 2013. *We still need to talk.* [online] Available at: <https://www.mind.org.uk/media-a/4248/we-still-need-to-talk_report.pdf> [Accessed 10 September 2021]

Mohamed, I. and Patel, D., 2015, April. Android vs iOS security: A comparative study. In *2015 12th International Conference on Information Technology-New Generations,* p.725-730. IEEE.

Mohlman, J., 2012. A community based survey of older adults' preferences for treatment of anxiety. *Psychology and Aging*, *27*(4), p.1182.

Nakamura, J. and Csikszentmihalyi, M., 2014. The concept of flow. In *Flow and the foundations of positive psychology,* p.239-263. Springer, Dordrecht.

NHS. 2021. *Get help with anxiety, fear or panic*. [online] Available at: <https://www.nhs.uk/mental-health/feelings-symptoms-behaviours/feelings-and-symptoms/anxiety-fear-panic/> [Accessed 10 September 2021]

NHS. 2021. *Symptoms - Generalised anxiety disorder in adults.* [online] Available at:<https://www.nhs.uk/mental-health/conditions/generalised-anxiety-disorder/symptoms/> [Accessed 10 September 2021]

NHS. 2021. *Treatment - Generalised anxiety disorder in adults* [online] Available at: <https://www.nhs.uk/mental-health/conditions/generalised-anxiety-disorder/treatment/> [Accessed 10 September 2021]

NHS. 2021. *How do I check my pulse?* [online] Available at: <https://www.nhs.uk/common-health-questions/accidents-first-aid-and-treatments/how-do-i-check-my-pulse/#:~:text=What's%20a%20normal%20heart%20rate,40%20to%2060bpm%2C%20or%20lower.> [Accessed 16 September 2021]

Nidhra, S. and Dondeti, J., 2012. Black box and white box testing techniques-a literature review. *International Journal of Embedded Systems and Applications (IJESA)*, *2*(2), p.29-50.

Nyklíček, I. and Kuijpers, K.F., 2008. Effects of mindfulness-based stress reduction intervention on psychological well-being and quality of life: is increased mindfulness indeed the mechanism?. *Annals of behavioral medicine*, *35*(3), p.331-340.

Omylinska−Thurston, J., McMeekin, A., Walton, P. and Proctor, G., 2019. Clients' perceptions of unhelpful factors in CBT in IAPT serving a deprived area of the UK. *Counselling and psychotherapy research*, *19*(4), p.455-464.

Parsons, D., 2020. The Java Story. In *Foundational Java* p.1-10. Springer, Cham.

Perez, S. 2021. *App stores saw record 218 billion downloads in 2020, consumer spend of $143 billion* [online] Available at: <https://techcrunch.com/2021/01/13/app-stores-saw-record-218-billion-downloads-in-2020-consumer-spend-of-143-billion/> [Accessed 10 September 2021]

Poo, D., Kiong, D. and Ashok, S., 2007. *Object-oriented programming and Java*. Springer Science & Business Media.

PSYT Ltd. (2020). Anxiety Solution.

Que, P., Guo, X. and Zhu, M., 2016, December. A comprehensive comparison between hybrid and native app paradigms. In *2016 8th International Conference on Computational Intelligence and Communication Networks (CICN),* p.611-614. IEEE.

Ray, T.N., Franz, S.A., Jarrett, N.L. and Pickett, S.M., 2020. Nature enhanced meditation: effects on mindfulness, connectedness to nature, and pro-environmental behavior. *Environment and Behavior.*

Seah, M.L. and Cairns, P., 2008. From immersion to addiction in videogames. *People and Computers XXII Culture, Creativity, Interaction 22*, p.55-63.

Serrano, N., Hernantes, J. and Gallardo, G., 2013. Mobile web apps. *IEEE software*, *30*(5), p.22-27.

Sheikh, A.A., Ganai, P.T., Malik, N.A. and Dar, K.A., 2013. Smartphone: Android Vs IOS. *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, *1*(4), p.141-148.

Sin, D., Lawson, E. and Kannoorpatti, K., 2012, June. Mobile web apps-The non-programmer's alternative to native applications. In *2012 5th International Conference on Human System Interactions,* p.8-15. IEEE.

StatCounter. 2021. OS Market Share. [online] Available at: <https://gs.statcounter.com/os-market-share/mobile/worldwide> [Accessed 10 September 2021]

Sucala, M., Cuijpers, P., Muench, F., Cardoș, R., Soflau, R., Dobrean, A., Achimas−Cadariu, P. and David, D., 2017. Anxiety: There is an app for that. A systematic review of anxiety apps. *Depression and anxiety*, *34*(6), p.518-525.

Tun, P.M., 2014. Choosing a mobile application development approach. *ASEAN Journal of Management and Innovation*, *1*(1), p.69-74.

Watson, G.R. and DeBardeleben, N.A., 2006. Developing scientific applications using eclipse. *Computing in Science & Engineering*, *8*(4), p.50-61.

Web.archive.org. 2021. *First Android Application - Snake*. [online] Available at: <https://web.archive.org/web/20080511192655/http://www.mobiles2day.com/2007/11/14/first-Android-application-snake/> [Accessed 18 September 2021]

Williams, C. and Martinez, R., 2008. Increasing access to CBT: stepped care and CBT self-help models in practice. *Behavioral and cognitive psychotherapy*, *36*(6), p.675-683.

Wolfson, M. and Felker, D., 2013. *Android developer tools essentials: Android Studio to Zipalign*. " O'Reilly Media, Inc.".

Yener, M. and Dundar, O., 2016. *Expert Android Studio*. John Wiley & Sons.

Yu, B., Hu, J., Funk, M. and Feijs, L., 2016. A study on user acceptance of different auditory content for relaxation. In *Proceedings of the Audio Mostly 2016,* p.69-76.

Zavarize, S.F., Paschoal, M.A. and Wechsler, S.M., 2016. Effects of physiotherapy associated to virtual games in pain perception and heart rate variability in cases of low back pain. *Manual Therapy, Posturology & Rehabilitation Journal*, *14*, p.1-7.

Appendix A - Requirements

Functional requirements

1. Over all application
    - 1.1. The system must provide information about mental health care available to users
        - 1.1.1. The system must provide links to outside mental health resources
    - 1.2. The system will not require users to register
    - 1.3 .The system must launch on the home screen
        - 1.3.1. The system must provide a way to navigate to each activity from the home screen
    - 1.3. The system must provide a way to return to the home screen on every activity
    - 1.4 . The system must provide a heading on each activity, informing the user of the current activity
    - 1.5. The system must provide information regarding the application
    - 1.6. The system must at least provide 4 different activities

2. Breathe activity
    - 2.1. The system must show a looping graphic of an expanding and collapsing shape to assist users with deep calm breathing
    - 2.2. The graphic should be a 3-6 second inhale/ exhale
    - 2.3. The graphic must constantly loop until the user exits the page
    - 2.4. The system must provide instructions for the activity

3. Sounds activity
    - 3.1. The system will have a variety of calming sounds for user to choose between
    - 3.2. The system must show each sound option on the screen
    - 3.3. The system must label each sound with a clear description
    - 3.4. The system must provide a way to play each sound
    - 3.5. The system must provide a way to pause each sound
    - 3.6. The system must visually communicate to the user which sound is playing
    - 3.7. The system should only play one sound at once.
        - 3.7.1. When a user attempts to play a second sound, the system must first stop all other sounds from playing
    - 3.8. The system must stop any playing sound upon exit of the activity

4. Blocks activity
    - 4.1. The system must clearly show the target block
        - 4.1.1. The system must give a randomised sequence of target blocks
        - 4.1.2. The system must only show one target block at any given time

4.1.3. The system must show a new target block after the user selects the current target

4.2. The system must provide at least 3 difficulty levels

4.2.1. The system must provide a means of selecting these difficulties

4.2.2. The three difficulty levels must clearly vary

4.3 The system must reward a user with a point for each correctly identified target block

4.4 The system must show a score counter

4.5. The system must have a reset option for the counter always visible

4.6.1 the system must reset the score to 0 each time the reset option is selected

4.6. The system must reset the counter each time the activity ends

4.7. The system must provide instructions for the activity

5. Maze activity

5.1. The system must show continuously moving obstacles for the user to navigate through

5.1.1. The system must randomise the position of the obstacles

5.1.2. The system must provide a way through each obstacle

5.2. The system must provide at least 3 difficulty levels

5.2.1. The system must provide a means of selecting these difficulties

5.2.2. The three difficulty levels must clearly vary

5.3. The system must end the game if the player interacts with an obstacle

5.4. The system must provide a way for the user to easily reset the game

5.5. The system must have a point counter for each obstacle successfully manoeuvred

5.5.1. The system must reset the counter for each run of the game

5.6. The system must provide a way for the user to control the player

5.7. The system must provide instructions for the activity

Non-functional requirements

1. Performance

1.1. The system must load the application within 3 seconds

1.2 The system must load each activity within 2 second

1.3. The system must respond to user inputs within 0.5 seconds

1.4. The system must maintain an average of 30 frames per second

1.5. The system must have a maximum of 1% skipped frames per session

2. Space

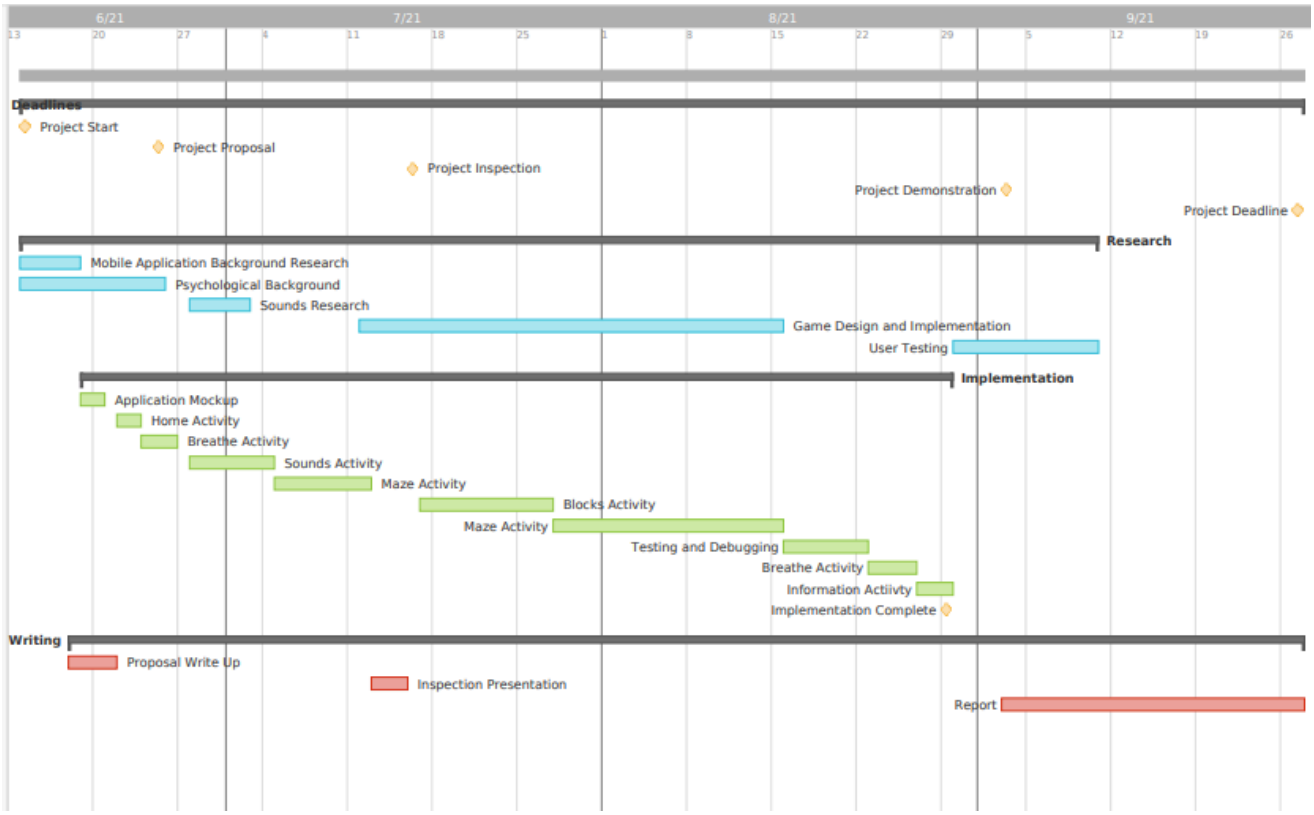
2.1. The System must not exceed 20MB

3. Portability
    3.1. The system must be compatible with Android mobile devices
    3.2. The system must be compatible with Android emulators
    3.3. The system must ensure its usability with future Android devices
4. Reliability
    4.1. The system must operate without crashing
    4.2. The system must maintain a mean time between failures of 150 hours
5. Usability
    5.1. The system must be intuitive and easy to use
    5.2. The system should be understood by the user within 30 seconds
    5.3. The system must scale to the different screen sizes of Android devices
    5.4. The system must have a consistent layout for each activity
    5.5. The system must have a consistent colour scheme
    5.6. The system text size and colour must be easily readable
    5.7. The system must use the same language throughout
6. Maintainability
    6.1. The system code should be well documented
    6.2. The system code must be easy to read and change

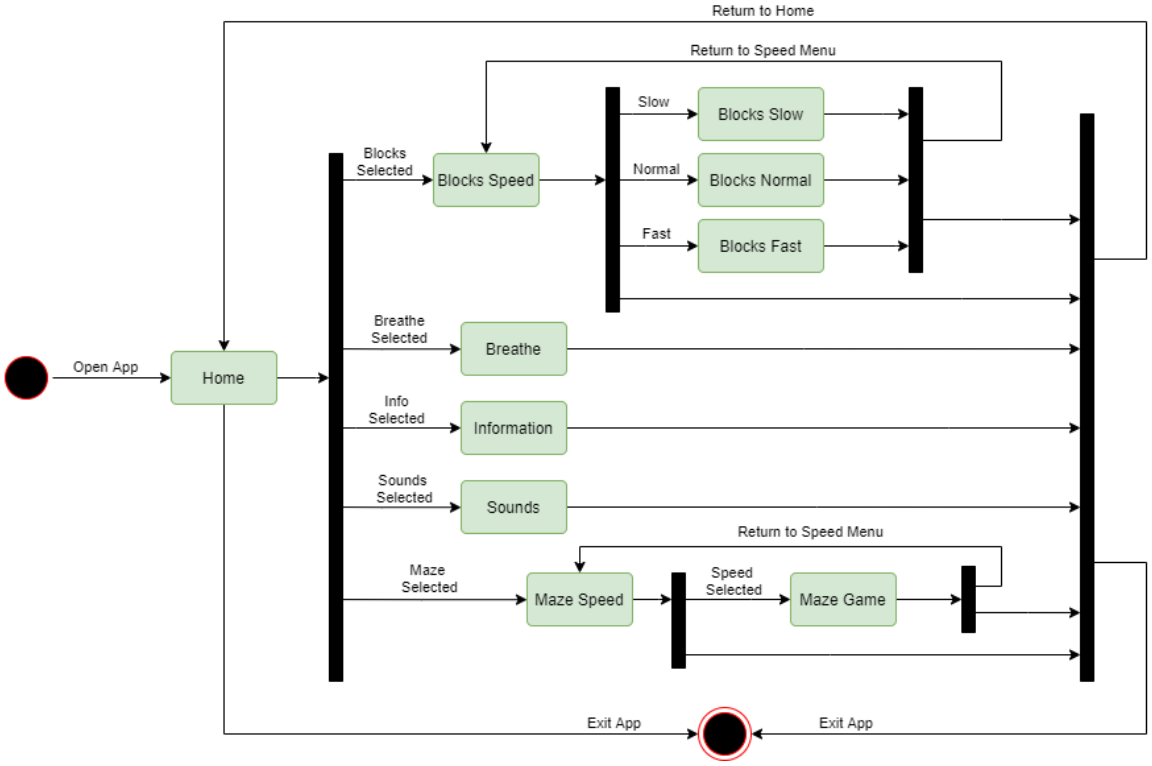
7. Scaleability
    7.1. The system must be developed in a way that new activities can be easily added, without affecting existing functionalities
    7.2. The system must be developed to be easily adapted for other mobile operating systems

8. Implementation
    8.1. The system must be developed using Android studio
    8.2. The system code must be developed using Java
    8.3. The application must not require any other external dependancies other than what has been included within the system.

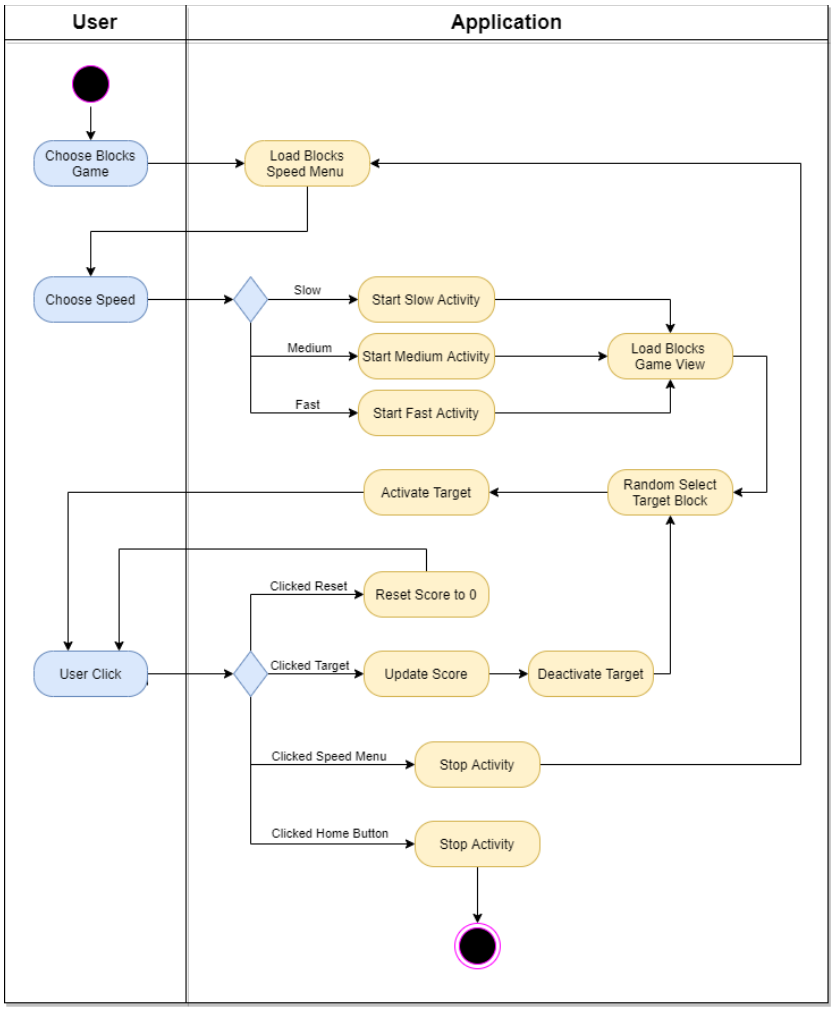
Appendix B - Gantt Chart



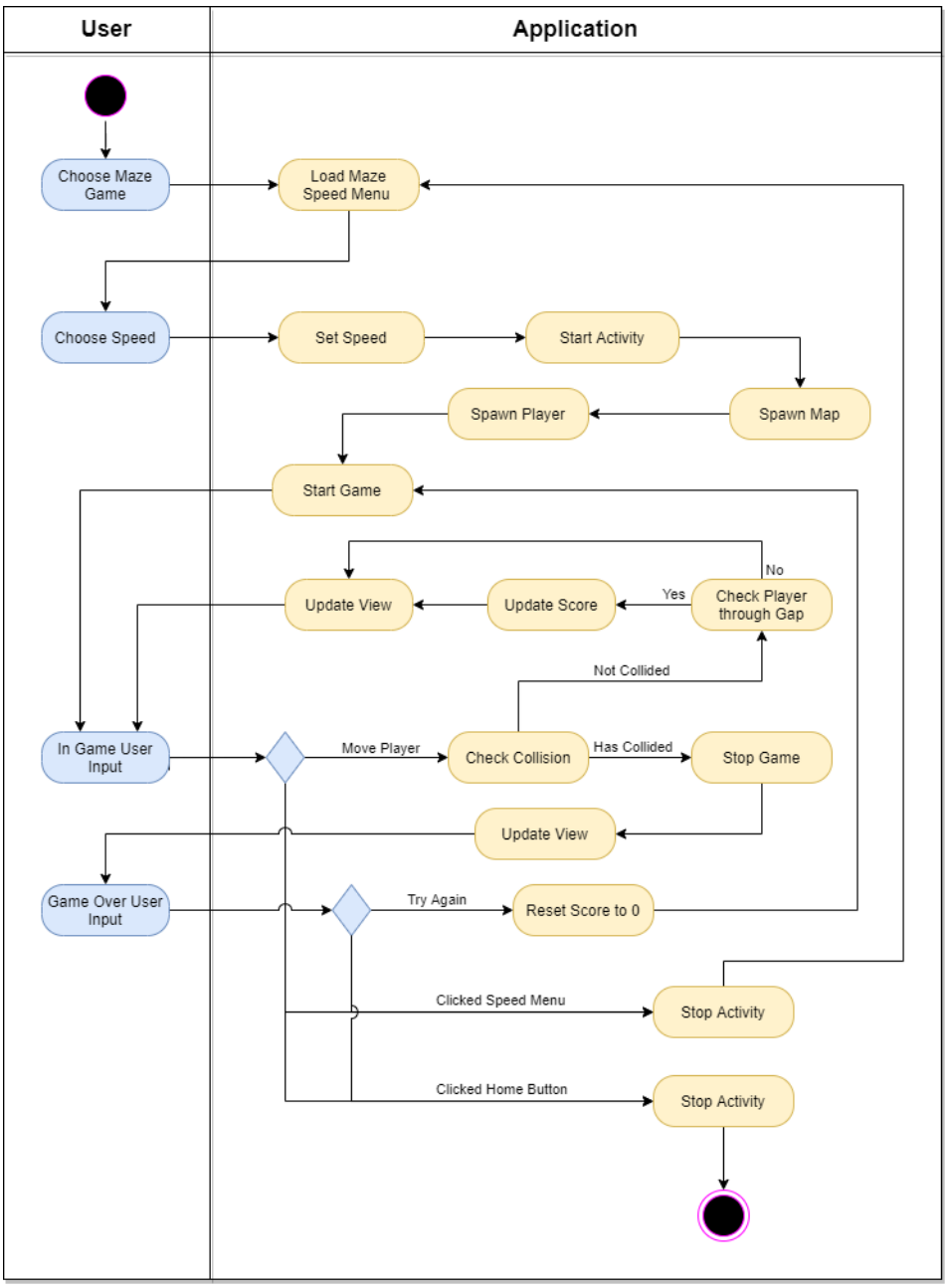
Appendix C - UML Diagrams



State Diagram of Activities

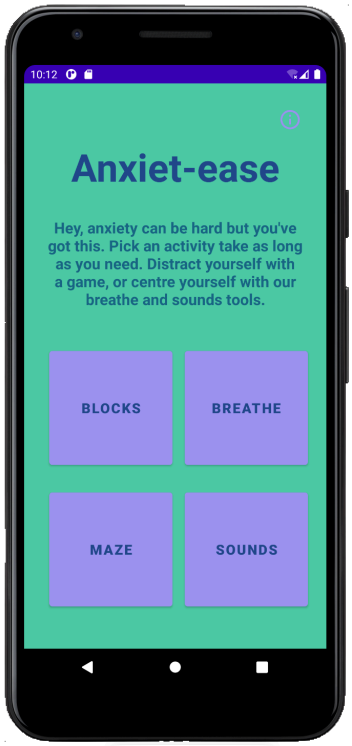


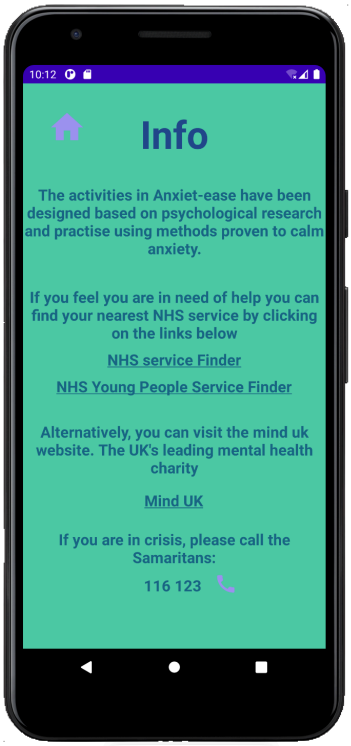
Activity Diagram of Blocks Game

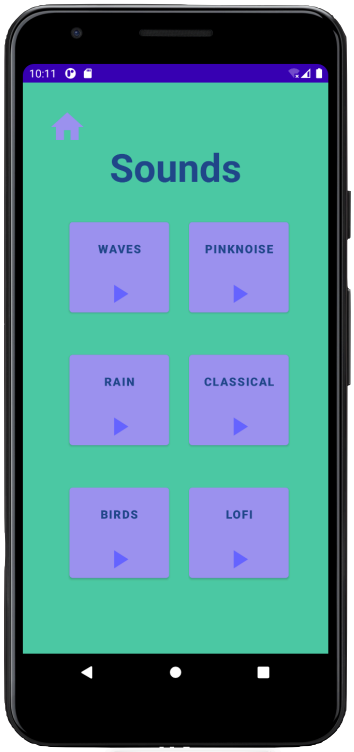


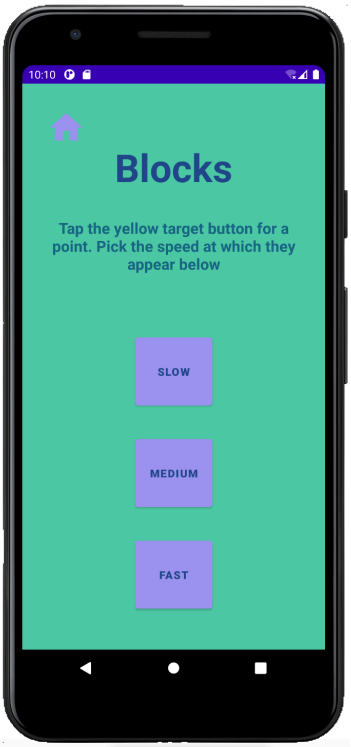
Activity Diagram of the Maze Game

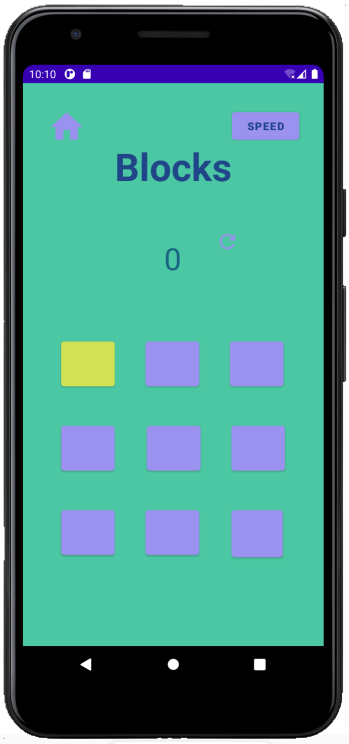
Appendix D - Application Screenshots

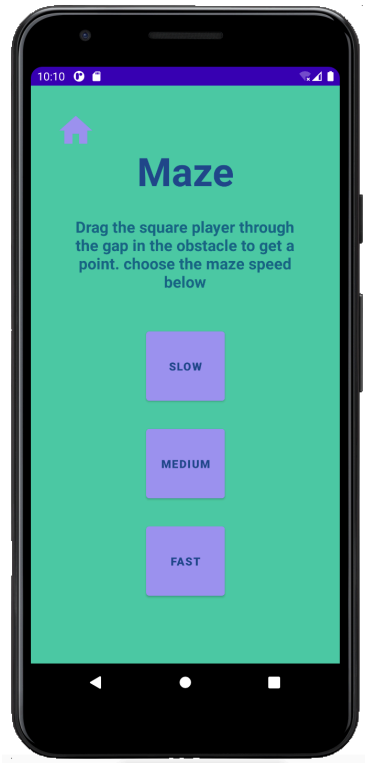


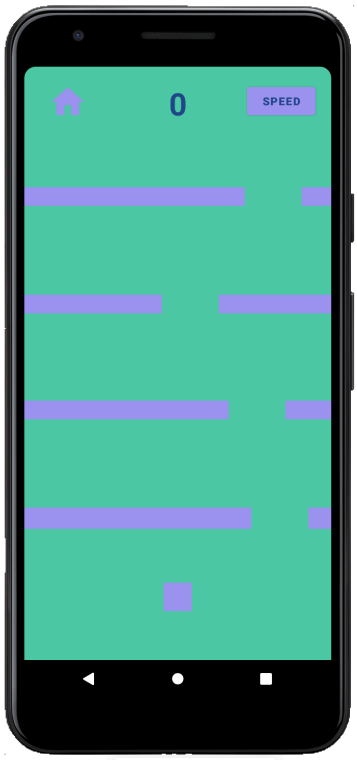


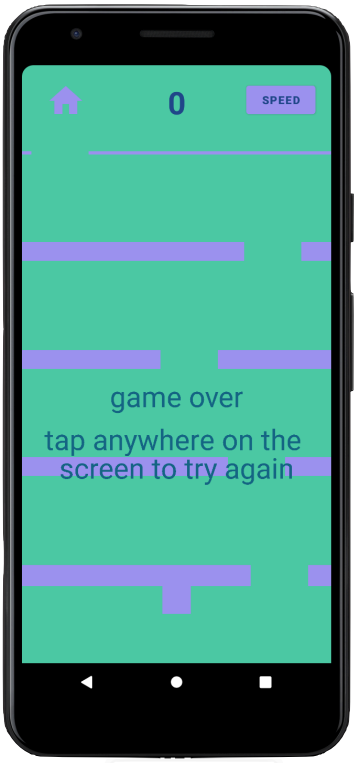







10:10
Maze
Drag the square player through the gap in the obstacle to get a point. choose the maze speed below
SLOW
MEDIUM
FAST


0
SPEED


0
SPEED
game over
tap anywhere on the screen to try again

Appendix E - Test Cases

ID	Test	Expected Outcome	Pass / Fail
Home Activity			
1	Application launch	Application opens to the home screen	**Pass**
2	User clicks the Breathe button	Breathe activity starts	**Pass**
3	User clicks the Sounds button	Sounds activity starts	**Pass**
4	User clicks the Blocks button	Blocks speed menu shows	**Pass**
5	User clicks the Maze button	Maze speed menu shows	**Pass**
6	User clicks the Information button	Information activity starts	**Pass**
Information activity			
7	Activity launch	The information activity will launch, displaying additional information	**Pass**
8	User clicks NHS Service finder link	Devices internet browser opens, loading the NHS service finder website	**Pass**
9	User clicks the youth NHS service finder link	Devices internet browser opens, loading the NHS youth service finder website	**Pass**
10	User clicks mind uk link	Devices internet browser opens, loading the Mind UK website	**Pass**
11	User clicks the Samaritans helpline phone	Devices phone opens, with the number entered ready to click dial	**Pass**
Breathe activity			
12	Activity launch	The animation automatically starts, looping continuously until the user exits the activity	**Pass**
13	User clicks Home button	The breathe activity finishes, returning the user to the home screen	**Pass**
Sounds Activity			
14	Activity launch	The sounds activity will open, displaying the 6 sounds and a home button	**Pass**
15	User plays waves sound	Waves sound plays, play image on button changes to pause	**Pass**
16	User pauses waves sound	Waves sound pauses, pause image on button changes to play	**Pass**
17	User plays pink noise sound	Pink Noise sound plays, play image on button changes to pause	**Pass**
18	User pauses pink noise sound	Pink Noise sound pauses, pause image on button changes to play	**Pass**
19	User plays rain sound	Rain sound plays, play image on button changes to pause	**Pass**

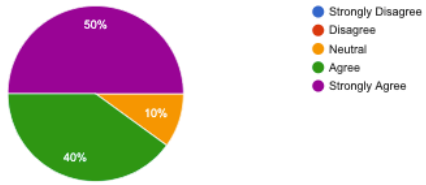
ID	Test	Expected Outcome	Pass / Fail
20	User pauses rain sound	Rain sound pauses, pause image on button changes to play	**Pass**
21	User plays classical sound	Classical sound plays, play image on button changes to pause	**Pass**
22	User pauses classical sound	Classical sound pauses, pause image on button changes to play	**Pass**
23	User plays birds sound	Birds sound plays, play image on button changes to pause	**Pass**
24	User pauses birds sound	Birds sound pauses, pause image on button changes to play	**Pass**
25	User plays lofi sound	Lofi sound plays, play image on button changes to pause	**Pass**
26	User pauses lofi sound	Lofi sound pauses, pause image on button changes to play	**Pass**
27	User plays a sound, while another is already playing	The old sound pauses and the pause image on button changes to play. The sound clicked will then play	**Pass**
28	User clicks Home button	Any sounds playing will stop. The sounds activity finshes and the home activity will start	**Pass**
Blocks Activity- speed menu			
29	Activity launch	Speed menu opens containing instructions for the game and buttons for each level	**Pass**
30	User clicks Slow speed	Blocks game launches on the slow speed	**Pass**
31	User Clicks Medium speed	Blocks game launches on the medium speed	**Pass**
32	User clicks Fast speed	Blocks game launches on the fast speed	**Pass**
Blocks Activity- Slow level			
33	User clicks the target button	A point is awarded. The target deactivates. A new target is lit up after 5 seconds	**Pass**
34	User clicks a button that is not a target	No changes to game	**Pass**
35	User clicks the reset button	The score returns to 0	**Pass**
36	User clicks the home button	The blocks game finishes and the home activity will start	**Pass**
37	User clicks the speed menu	The blocks game finishes, the user is taken to the speed menu.	**Pass**
Blocks Activity- Medium level			
38	User clicks the target button	A point is awarded. The target deactivates. A new target is immediately activated.	**Pass**
39	User clicks a button that is not a target	No changes to game	**Pass**
40	User clicks the reset button	The score returns to 0	**Pass**

ID	Test	Expected Outcome	Pass / Fail
41	User clicks the home button	The blocks game finishes and the home activity will start	**Pass**
42	User clicks the speed menu	The blocks game finishes, the user is taken to the speed menu.	**Pass**
Blocks Activity- Fast level			
43	User clicks the target button	A point is awarded. The target deactivates. A new target is activated after 2 seconds	**Pass**
44	User does not click the target button	The target button will deactivate after 1 second. A new target is activated after 2 seconds	**Pass**
45	User clicks a button that is not a target	No changes to game	**Pass**
46	User clicks the reset button	The score returns to 0	**Pass**
47	User clicks the home button	The blocks game finishes and the home activity will start	**Pass**
48	User clicks the speed menu	The blocks game finishes, the user is taken to the speed menu.	**Pass**
Maze activity			
49	Activity launch	Speed menu opens containing instructions for the game and buttons for each level	**Pass**
50	User clicks Slow speed	The maze game starts, the obsticles generating at the slow speed from the top of the screen	**Pass**
51	User Clicks Medium speed	The maze game starts, the obsticles generating at the medium speed	**Pass**
52	User clicks Fast speed	The maze game starts, the obsticles generating at the fast speed	**Pass**
53	User drags player	Player is moved with user input	**Pass**
54	User navigates player through obstacle gap	A point is added to the score at the top of the screen	**Pass**
55	Player collides with wall	Game over screen shows	**Pass**
56	User clicks game over screen	Game resets	**Pass**
57	User clicks the home button	The maze game finishes, and the home activity starts	**Pass**
58	User clicks the speed menu	The maze game finishes, and the speed menu starts	**Pass**

Appendix F - User Evaluation

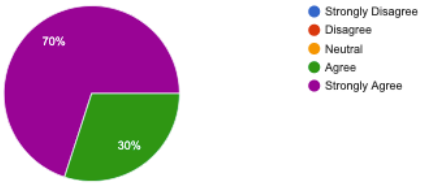
Anxiet-ease has had a positive effect on my anxiety

10 responses



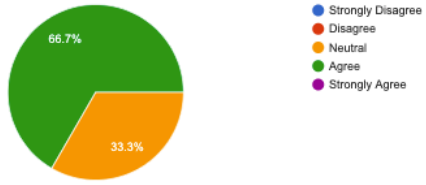
Anxiet-ease is easy to use

10 responses



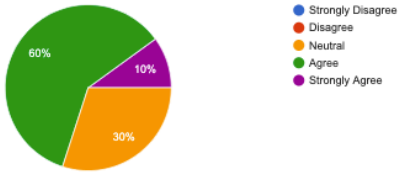
The breathe activity helped me regulate my breathing

9 responses



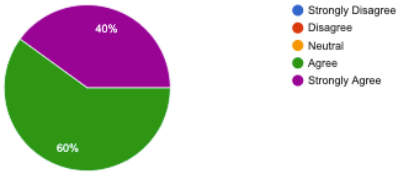
The sounds activity helped me to centre and calm myself

10 responses



The blocks activity provided me with a good distraction from my anxiety

10 responses

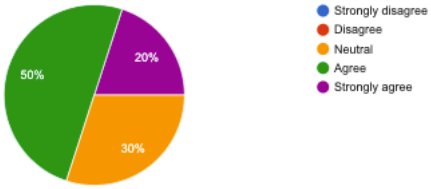


The maze activity provided a good distraction from my anxiety

10 responses

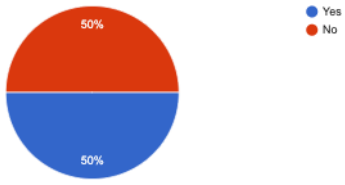
The games had a good range of difficulties for my skill level

10 responses



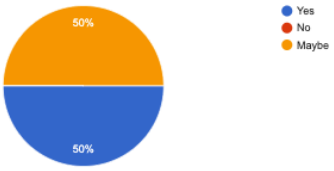
Have you ever used a mental health app before?

10 responses



Would you choose Anxiet-ease over other mental health apps?

10 responses



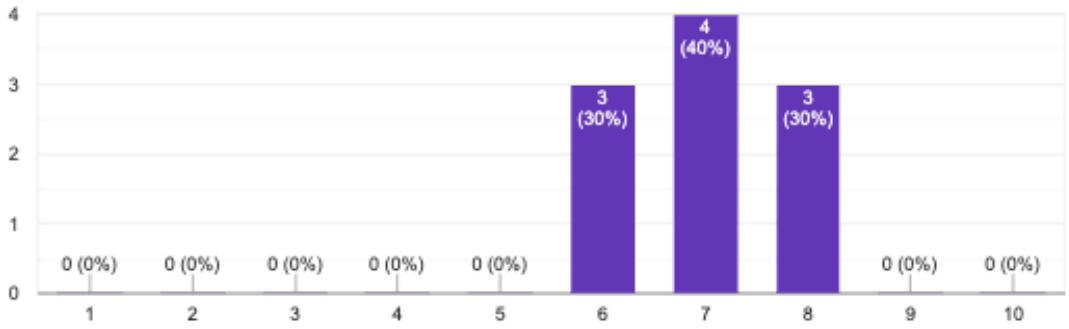
Please give a reason for your previous answer

7 responses

never used one before so wouldnt know
the gaming element
its really different from apps ive used before
mental health applications have never appealed to me before using anxietease
would have to use another app before saying yes
would use again for the games, apple watch has a better breathe function though
the variety

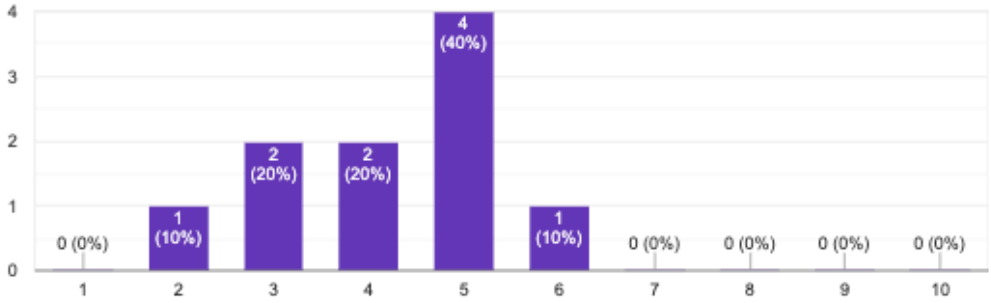
Please rate average anxiety level before using Anxiet-ease

10 responses



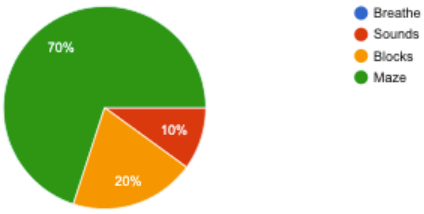
Please rate average anxiety level after using Anxiet-ease

10 responses



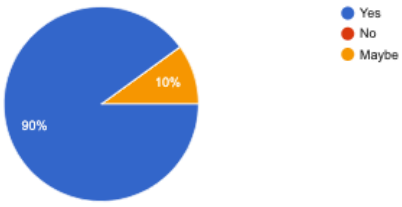
Which activity was your favourite ?

10 responses



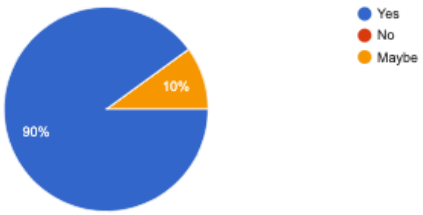
Would you use Anxiet-ease again?

10 responses



Would you recommend Anxiet-ease to a friend/ family member ?

10 responses



What is your overall thoughts of Anxiet-ease? Any thoughts, comments and/or concerns?

10 responses

i really liked the maze game, it kept my attention so i didnt think about why i was anxious

ames were really good and unique, its not like other apps that can be abit boring

really like the mindfullness activities but the games were good

ames were good distractions but i felt just as anxious after playing them

ry different from other mental health apps, i like the variety of choices

the speed levels in the blocks could be better, the fat mode isnt that fast

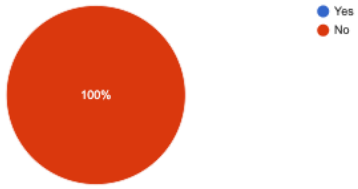
activities in the app would make it even better, the choice is a little bit limited as it is

istraction games are a really good idea and worked well for me

y liked the selection of differnt sounds the lofi sound was really calming to focus on

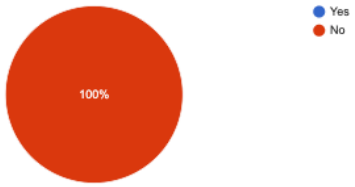
Did anxiet-ease crash at any point during use?

10 responses



Did you experience any lag or glitches when using anixet-ease?

10 responses



If you answered yes to any of the questions in this section please give details below.

0 responses

No responses yet for this question.

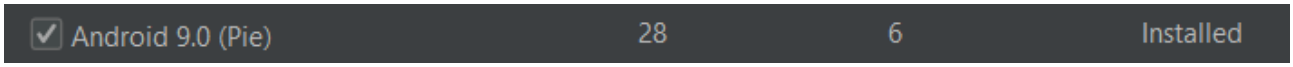
Appendix G - Github

https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2020/sxs1805

Running the application on Desktop using Android Studio:

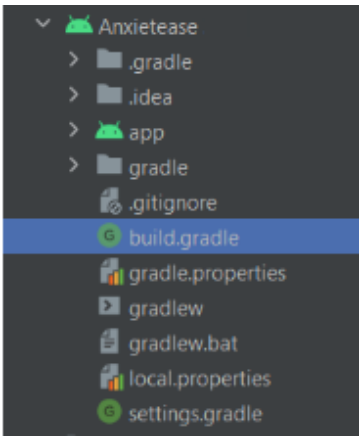
Prerequisites:

- Android Studio is downloaded and installed
- Ensure the “Android 9.0 (Pie)” SDK is installed and active. If not installed during installation of Android Studio, go to Tool -> SDK Manager and install it there (see image below).



Running:

1. Click the link to the repository
2. Download the zip file and extract file in desired location
3. On Android Studio, go to File -> Open and open the project by clicking on the build.gradle file within the gradle folder of the project (see image below)



4. The gradle may take up to a couple of minutes to build
5. Make a virtual device from Android Studio by going to Tools -> AVD Manager, clicking “Create Virtual Device” on the bottom left of the window, and cloning the Google Pixel 3a (see image below)

6. Run the project ensuring the Pixel 3a is selected as the device. The emulator should pop-up in a new window. It will take a little time for the emulator to load the application.
7. Once loaded, enjoy the application

Note: If you would like to run Anxiet-ease on your own Android device, connect the device to your computer, and select it in the AVD menu.