**UNIVERSITY OF HERTFORDSHIRE**

**Department of Computer Science**

**Modular BSc Honours in Computer Science**

**6COM2018 - Computer Science Project**

**Final Report**

**March 2024**

**TITLE OF PROJECT**

**Using software development principles and practices to develop software to effectively manage addictions to software by employing behavioural modification techniques.**

**Author's initials and surname**

**M. Khidr**

**Supervised by: Stillianos Vidalis**

Contents

[List of Figures: 3](#_Toc160443232)

[List of Tables: 3](#_Toc160443233)

[Abstract: 4](#_Toc160443234)

[1.0 Introduction: 5](#_Toc160443235)

[1.1 Project Background: 5](#_Toc160443236)

[1.2 Project Motivation: 5](#_Toc160443237)

[1.3 Project Hypothesis: 5](#_Toc160443238)

[1.4 Project Aims and Objectives: 5](#_Toc160443239)

[1.5 Report expectations: 6](#_Toc160443240)

[2.0 Literature Review: 6](#_Toc160443241)

[2.1 Addiction and the Brain: 7](#_Toc160443242)

[2.2 The Psychology of Addictions: 7](#_Toc160443243)

[2.3 Understanding Technology Addictions: 8](#_Toc160443244)

[2.4 The Consequences of Desktop Software Addictions: 10](#_Toc160443245)

[2.5 Behavioural Treatments: 11](#_Toc160443246)

[2.6 Review of similar solutions: 12](#_Toc160443247)

[2.7 Considerations: 14](#_Toc160443248)

[3.0 Methodology: 15](#_Toc160443249)

[3.0.1 Risk Assessment: 15](#_Toc160443250)

[3.0.2 Design Specification: 16](#_Toc160443251)

[3.0.3 Data Flow Diagram: 16](#_Toc160443252)

[3.1 Resource Analysis and Justification: 18](#_Toc160443253)

[3.1.1 Methodology: 19](#_Toc160443254)

[3.2 Back End Development 19](#_Toc160443255)

[3.2.1 Selecting the Application: 20](#_Toc160443256)

[3.2.2 Timing the Applications: 22](#_Toc160443257)

[3.2.3 Terminating Applications: 22](#_Toc160443258)

[3.2.4 Blocking Applications: 23](#_Toc160443259)

[3.3 Front End Development 24](#_Toc160443260)

[3.3.1 Prototype Implementation: 25](#_Toc160443261)

[3.3.2 Blocker Mode: 27](#_Toc160443262)

[3.3.3 Persistence: 29](#_Toc160443263)

[3.4 Challenges Faced: 29](#_Toc160443264)

[4.0 Testing and Evaluation: 31](#_Toc160443265)

[4.1 Testing Methodology: 31](#_Toc160443266)

[4.2 Evaluation Methodology: 37](#_Toc160443267)

[4.3 Evaluation: 37](#_Toc160443268)

[5.0 Conclusion 40](#_Toc160443269)

[5.1 Future Work: 40](#_Toc160443270)

[6.0 Project Management Review 41](#_Toc160443271)

[References: 44](#_Toc160443272)

[Bibliography: 46](#_Toc160443273)

[Appendix A: Initial Gantt Chart 46](#_Toc160443274)

[Appendix B: Revised Gantt Chart 47](#_Toc160443275)

[Appendix C: Testing Table 48](#_Toc160443276)

# List of Figures:

* Figure 1: Off the Grid System Blocker ………………………………………………………………………………14
* Figure 2: Freedom Subscription Fee ………………………………………………………………………………….15
* Figure 3: Freedom Customization …………………………………………………………………………………….15
* Figure 4: Use case Diagram ……………………………………………………………………………………………….20
* Figure 5: Displaying the Code for Open File Function ………………………………………………………..21
* Figure 6: Screenshot of File-menu Selection ……………………………………………………………………..22
* Figure 7: Popup of Success ……………………………………………………………………………………………….22
* Figure 8: Popup when selecting non .exe …………………………………………………………………………..23
* Figure 9: Code of the Timing Function……………………………………………………………………………….23
* Figure 10: Timer Termination Code ………………………………………………………24
* Figure 11: Code of the App Blocker …………………… 25
* Figure 12: GUI Prototype …... 26
* Figure 13: Screenshot of the finished main GUI. …………… 27
* Figure 14: Code for GUI ………………………. 28
* Figure 15: The options-menu…. 29
* Figure 16: Blocker mode GUI……... 29
* Figure 17: Code snippet of the While loop blocker ………. 30
* Figure 18: snippet of the revised recursion app blocker. ……… 31
* Figure 19: Showcasing Test case 2 Scenario of invalid entry. ……….. 34
* Figure 20: Showcasing Test case 1 …………………. 34
* Figure 21: Showcasing Test Case 4. ……………………………. 35
* Figure 22: Showcasing Test cases 3 and 4. …………………….35.
* Figure 23: Showcasing the Result of Test case 4. ………………………36
* Figure 24: Result of Test case 7………………………………………………….36
* Figure 25: Test case 6, clearing Timelist ………………37.
* Figure 26: screenshot showing Sekiro play time. 39
* Figure 27: the date of last played …40.
* Figure 28: Gantt chart 1 ……………. 42
* Figure 29: revised Gantt chart …………...43
* Figure 30: Github Repo ……………….44

# List of Tables:

* Table 1: Development Risk Assessment ……………………16
* Table 2: Subset of the Test cases ………….32
* Table 3: Evidence …………………………….38

# Abstract:

Addiction to technology is the most prevalent addiction plaguing society today; one such technology being desktop software such as video games, the Internet, and TV streaming apps. The reason why it is so common for individuals to be hooked on technology is due to the lack of entry barriers to technologies themselves, the lack of formal diagnosis of the addiction and because of society’s normalization of addictive behaviour, therefore it is the addiction with the greatest potential to affect the majority of people and more so it is the only addiction that is incredibly likely to be afflicted on children and adolescents. This report analyses addictions, their causes, and the consequences that technology addictions have on individuals. Furthermore, this report highlights the complexities of treating addictions and why classical medication is ultimately ineffective and promotes an alternative treatment, behavioural modification techniques. The behavioural modification techniques were implemented via a software application to use technology to treat technology addictions via behavioural techniques such as: detoxification and tapering off to wean addicts off their addiction by giving them control over it. As an individual who also exhibited an addiction to Sekiro: Shadows die twice (A videogame); the test to see if the developed solution was effective was for it to be able to combat my own personal addiction, and as a result through using the application over the winter break as a testing period I was able to eventually give up my addiction, dedicate more hours to my assignments as well as other activities such as going to the gym and socializing.

# 1.0 Introduction:

## 1.1 Project Background:

The Oxford dictionary (2020) characterizes an addiction as “the condition of being unable to stop using or doing something as a habit for a prolonged period of time, especially something harmful”, therefore a software addiction is a long term, compulsive, obsessive need to utilise software with or without consciously choosing to do so whilst in the pursuit of gratification. The scope of this gratification is wide, one common desire is needing to utilise software such as social media as a method of passing time whilst performing mundane activities such as commuting to work, another is playing video games to attain a dopamine rush. Performing one of these activities occasionally would be deemed normal and safe, but regular repetition daily for an extended period and or duration is deemed an addiction. Unfortunately, due to the wide scale availability of modern technology along with current societal norms, many people are unconsciously suffering from software addictions.

## 1.2 Project Motivation:

The inspiration behind this project is an epiphany I had whilst on my commute to university one morning, as I stood aboard the train I glanced around and noticed that without fail every person in this carriage(myself included) was utilising a piece of technology; be it a tablet, a phone or a laptop and that people most likely have done so unconsciously out of repetitive habitual instinct for wanting to pass time until they reach their destination.

## 1.3 Project Hypothesis:

Specifically for this project I will be targeting desktop software addictions because Mobile software addiction has been tackled both by manufacturers and independent developers; both closed and open-source implementations. The hypothesis for this project is: By utilizing Software development principles, practices, and methodologies to devise an automated software solution which implements behavioural reinforcement techniques, it will reduce the addictive compulsion of utilizing addictive software in an effective, long term and most importantly safe way.

## 1.4 Project Aims and Objectives:

The aim for this project is to have a functional application that will implement behaviour modification techniques to correctly and effectively to treat addictions to desktop software applications. The development objectives that I plan to achieve are:

* The application should be modern and aesthetically pleasing to use.
* The application should be intuitively designed to promote usability and accessibility.
* The application should be easy to navigate and interact with.
* The application should be cross platform to run on multiple desktop operating systems, Linux, Windows, Mac OS.
* The application should integrate a menu system which offers additional functionality such as a tutorial page and documentation and additional features.
* The application should implement persistency so that the user can make changes and alterations that can be saved to enhance the applications quality of life.

The research objectives that I plan to complete are:

* Furthering my understanding and ability to implement the principles and practices of application-based software development.
* To discover and familiarise myself with tools for project development and management.
* To discover the tools available to create desktop applications.
* To understand the components of addiction.
* To understand the causes, consequences, and treatments of various addictions.

The evaluation objective of the project with regards to determining if the project is successful or not is to test over a given period whether the usage of addictive applications has decreased exponentially in the test subject over the duration of the evaluation period, if it has then that would deem the project to be a major success, if there is slight reduction in usage the project can be considered a partial success, and if there is no change in the frequency that the addictive applications are used then the project is deemed unsuccessful. Additional personal objectives that I would like to achieve are improving my time and project management skills, I aim to improve my programming and testing ability as these are highly sought skills in the tech industry.

## 1.5 Report expectations:

Section 2 will focus on the Literature review process which deals with researching topics that are relevant to the subject matter, sourcing information that is both pertinent and nuanced regarding addiction and software development, furthermore this section will evaluate similar solutions to the problem that are already available.

Section 3 will detail the projects methodology and will delve into the approach and the decisions that have been made, expanding upon the successes, the failures and the justifications and explanations for why an approach taken.

Section 4 will discuss the Testing and evaluation process of the application; detailing the testing methodology, the test cases, and discussing the testing results, in this section the efficacy of the solution produced for the problem will be scrutinized to see if the evaluation objective is met, how many of the development objectives were met, the flaws of the solution will be highlighted and reasoned for.

In Section 5 the points made in the evaluation of the report will be analysed and reflected upon to see if the initial aim of the project has been met, as well as giving recommendations on what can be done in the future to revise and improve the project as well as discussing the knowledge and skills that were developed over the course of the project.

Section 6 will review how the project was managed and will delve into the tools and strategies that were used to manage the project.

# 2.0 Literature Review:

## 2.1 Addiction and the Brain:

The American society of addiction medicine defines addiction to be “a primary, chronic disease of brain reward, motivation, memory, and related circuitry.” (ASAM, 2019) This is the medical definition of addiction which highlights that it is a self-inflicted voluntary ailment and is understood as the cyclic and constant intention to perform actions which provide gratification through the release of pleasure hormones such as dopamine, endorphins, and adrenaline. This model highlights that the user makes intentional choices for the purpose of gratification via hormone secretion.

When an individual becomes addicted to desktop software such as video games, over time their brain becomes tolerant to the activity to which the addicted party must increase their consumption to feel a “high”. Discerned by Wein (2017) If there is no intervention, the pleasure/reward circuits in the brain compel the addicted individual to constantly seek for more and more which eventually results in damage to regions of the brain and subsequently a decline in mental acuity.

## 2.2 The Psychology of Addictions:

Put forth by Heather’s article in the British psychological society journal, “addictive behaviour is completely involuntary and against the will of the person: addicts do not ‘use’ because they choose to, but because they are compelled to.” (Heather, 2017, p.3)

Modern scientists and psychologists put forth this argument that Addiction is not a choice as there is an absence of free will. Addiction and fulfilling addictions do not happen consciously but rather subconsciously and it is a culmination of reflexes that compels a person to perform the task to whit they are addicted to. In the context of the problem domain the afflicted party does not actively choose to use the desktop software that they are addicted to, but rather it happens unconsciously. Henceforth they are not able to stop nor control their usage because they never made the conscious decision to use it in the first place after becoming addicted.

This is in direct contrast to the earlier mentioned medical definition of addictions where it is all a choice and that the addicted party can choose not to, and therefore there is the moral implication that the addicted party must help themselves by choosing differently, the moral argument was put forth by psychologists in the 18th century and many psychologists still hold this position to be true due to all actions being seen as a choice being made. (Vohs and Baumeister, 2009).

The crucial flaw of the first position is that the addicted party is free of all responsibility and cannot be held accountable which poses a problem for addictions that are harmful to others such as reckless driving due to substance abuse as there are legal and ethical ramifications for some addictions which this position irrationally discards, some addictions are definitely more of a choice than others especially those that affect members of society, therefore this argument is centred around a deterministic viewpoint which for addictions that are solely self-inflicting harm could be argued for, but as a general argument it falls short due to a lack of recognition that all actions have an impact on wider society and it’s in this view that the argument can be classed as naïve. (Hoffman and Goldfrank, 1990).

However, the second position completely fails to account for Vohs and Baumeister’s counterpoint “When willpower has been depleted (such as by other acts of self-control, or even by decision making in any context); the likelihood of choosing the immediate pleasure increases”. (Vohs and Baumeister, 2009, p.231). This counter to the moral argument highlights how addiction causes a deterioration of the addicted parties mental health and ability to rationalise to the extent that their cognition of their actions is severely hampered and that the only factor considered for their actions is the utility and pleasure that can be gained, which is assumed to be via emotional impulses and not a logical assessment. This poses the question that if individuals are committing actions solely on emotional impulses due to a decline in mental faculty, is it just to hold them to the same standards as those who intentionally make choices whilst understanding potential consequences? To contextualise this, if a user is addicted to desktop streaming applications and as a result their physical health declines due to a sedentary lifestyle, assuming that they are not making the conscious choice to do so, is it their fault?

There isn’t a consensus on the blame of addictions, which is why a third argument has been put forth characterising addictions to be strictly an illness (also known as the Brain disease model of addiction). Heather understood that this illness initialises in the form of an initial choice to perform an activity, over time an unhealthy attachment and compulsion to repeatedly perform the activity develops because of poor choices, eventually resulting in free will and cognition being lost. This argument is a middle ground that resolves the flaws of the previous 2 arguments; an initial choice is made meaning that there is free will initially, but once the addiction has developed free will is lost and the afflicted party suffers from a disease, and at this point cannot be held accountable. (Heather 2017)

In evaluation the third argument of the brain disease model of addiction makes a collaboration of the first 2 arguments(the medical and the deterministic views) by both highlighting that addiction is caused by a free will initial choice, but later progresses to a deterministic compulsion to which the afflicted party has little to no choice in whether to perform the addicted activity or not, and therefore I would consider it to be an exhaustive understanding of the psychology of addictions whilst also providing a social insight of the severity of addictions and “is less stigmatizing than the view of addiction as a moral failing and brings hope that medications can be developed to address the disease” according to the (Butler Center for Research publication, 2021).

However, with recent developments in neuroimaging technology, some psychologists have argued that the BDMA (brain disease model of addiction) is not entirely conclusive given that the neurobiology of addicted patients is similar for a small subset of addicted individuals, but not concrete for a larger set of addicted individuals, hence the disparities have caused psychologists to rethink the model’s validity as it is deemed to be a “one dimensional portrayal”. (Heather *et al* 2018). Additionally, the BDMA cannot be extrapolated to explain the behaviour of animals in a laboratory that are exposed to an addictive substance because of the researchers having to rely on anthropomorphism which is a subjective interpretation. (Hall *et al* 2015) In conclusion, the BDMA is the most balanced and comprehensive understanding of addictions.

## 2.3 Understanding Technology Addictions:

Addictions to technology have always existed but did not become a mainstream phenomenon until the 20th century with the invention and widespread adoption of the Television set. The television was initially an uncommon good given the earliest models were incredibly expensive, however after the second world war there was a seismic shift in the delivery of public broadcasts and information from the radio to the Television. Thereafter the television set found its way into the homes of many and subsequently became the first modern technological addiction. Since the invention of the television set, we have seen an exponential increase in addictive technologies; from the desktop computer which has revolutionised the way we interact with the world, all the way back to the fruit machines of the 1980’s which captivated and enthralled adolescents in the UK.

After conducting interviews with adolescents afflicted with fruit machine addictions, Griffiths recognised that the addiction caused a decline of interpersonal skills and oft resulted in the degradation of communication skills, which can lead to the breakdown of relationships and potential feelings of social ostracization which contribute to poor mental health, a reduction in the addict’s employability as well as severe personality changes. Griffiths later compared the results of the interviews of fruit machine addicts with those afflicted with other technological, social and substance addictions and concluded that all addictions have 5 common components which are: (Griffiths, 1993 and 1995)

* Salience: when an activity becomes the most important thing of a person’s life and dominates their thoughts and feelings.
* Euphoria: the addiction provides a “buzz” or “high” feeling of enlightenment.
* Tolerance: where the activity must be amplified to give off a greater dose of the feelings mentioned. withdrawal symptoms; any ceasing of the activity for any duration results in states of uneasiness and or unpleasantness both physically and emotionally.
* Conflict: of interests where the addictive activity takes precedence over others such as work, school.
* Relapse: which refers to attempts at treating the addiction can be unsuccessful and the addicted party returns to said addiction and the state of addiction worsens.

What Griffiths (1993) discerned regarding fruit machine addictions can be extrapolated to any technology addiction be it hardware or software, Savci and Aysan also concluded that “the intensive use of technology is together with problematic or pathological consumption”. (Savci and Aysan, 2017, p.202) This quote highlights that excessive usage or attachment is the first sign of an addiction. Savci and Aysan also establish that “internet addiction can be likened to volatile substances” in the sense that both exploit the brains reward circuitry, by performing the activity repeatedly there is an increase in tolerance to dopamine and other pleasure hormones which drives the addiction higher hence greater usage is needed to release the pleasure hormones, both are commonly available, both have intense mental impacts and impairments and are incredibly to safely withdraw from. (Savci and Aysan 2017) It is well established that there is an inversely proportional relationship between Desktop software use and social connectedness; when desktop software usage is high through applications such as Video games, social connectedness plummets. Levounis and Sherer highlight and explain that the low entry barriers and lack of legislation surrounding usage of such technology will mean “Society’s dependence on addictive technologies will only increase”. (Levounis and Sherer, 2022, p.399)

## 2.4 The Consequences of Desktop Software Addictions:

Given that desktop software constitutes to a wide range of different things; listed below are a small subset of severe consequences of Desktop software addiction and a generalisation that is applicable to all desktop software:

Consequence 1: “withdrawal symptoms when not engaged include a diminishing social life”, identified by (Kumar and Mondal, 2018, p.61). Declining interpersonal skills and increased feelings of social ostracization, awkwardness, and anxiety; It has been established that there is an inversely proportional relationship between technology use and social connectedness such that an addiction to any technology (for example desktop video games) has detrimental results to one’s ability to form relationships, to socialise and greatly affects employability chances due to a decline in verbal communication skills. Furthermore, it leads to loneliness which often is a contributing factor to mental illnesses such as depression.

Consequence 2: Declining physical health. Spending copious amounts of time staring at screens is incredibly damaging to one’s eyesight leading to conditions such as myopia, sitting down for hours on end at a desk severely increases the risk of blood clots forming which are incredibly severe due to stopping circulation to the lower regions of the body and oft fatal, bad posture whilst seating can lead to chronic joint pains, replacing physical exercise with the addiction is noticeably going to lead to obesity, increased lethargy, and further health conditions synonymous with lack of exercise.

Consequence 3: Decline in workplace or academic performance. Cai and Tong (2022) established that addictions exploit regions of the brain such as the mesolimbic dopamine pathway; a region of the brain to which dopamine, a pleasure hormone is transported to and from the ventral tegmental area (VTA) which is responsible for knowledge acquisition, memory retention and other functions, in such way to which cognitive ability is directly hampered resulting in a lower quality of performance in say Exams, or in work tasks by hampering the ability of memory recollection and association due to weakening the activations from the frontal lobe and the parietal lobe. Additionally, gratification derived from success in academia or in a workplace is unlikely to stimulate the addicted individual due to being replaced by the addiction itself thereby causing. Tulubas, Karakose and Papadakis (2023) highlight that “Digital Addiction…. Is a significant factor influencing students’ academic achievement”.

Consequence 4: Significant personality change. All diseases have symptoms and the symptoms of addictions in general are the development of an obsessive-compulsive personality disorder (OCD). According to an article written by Mind.org (2019), Mind which is a mental health charity, explains OCD as “unwelcome, obtrusive thoughts, feelings, images, urges, worries, or doubts that keep coming into your mind. They may feel stuck in your mind, no matter what you do. You may worry what they mean or why they won't go away and feel very distressed by them. Compulsions are repetitive things that you do to reduce the distress or uncertainty caused by obsessions. Compulsions can be things you do physically, like repeatedly checking a door is locked. Or they can be things you do in your head, like repeating a specific word to yourself. Or they may involve others, such as asking people for reassurance”.

Consequence 5: Deterioration of sleep. Frequent usage of software be it on the Internet, the desktop or the mobile phone can lead to insomnia, poor sleeping patterns and poor quality of sleep. A study conducted by Mahmood, Hadad and Sayed (2022) on internet addiction and the sleep of university medical students found “About 81.62% of subjects who suffer from Internet addiction significantly had poor quality of sleep”. The cause of the decline in sleep is due to the emission of blue light from screens, blue light blocks the secretion of melatonin at night; a hormone that is responsible for inducing sleep, and as such this causes insomniac symptoms which has the side effects of sleeping less and sleeping worse. Sleep is a very important mechanism for the human body; it allows for cellular growth and repair to occur, helps with revitalization, and other important functions. The study found that the individuals with poor quality of sleep also suffered from lower levels of achievement as well as suffering from a worse demeanour.

## 2.5 Behavioural Treatments:

As established above addictions are incredibly complex with a multitude of symptoms and causes, as such there is no universal cure for any addiction. However, the methodology of treating software addictions is less fixated on medication courses but on behavioural modification techniques. Behavioural modification is the process of altering an individual’s thoughts, emotions, and their actions through the introduction of stimuli and feedback loops which force an individual to respond differently. Here are some examples of behavioural treatments to addiction:

Detoxification: to quote an explanatory article by the National institute of Health (2016) “detoxification is a set of interventions aimed at managing acute intoxication and withdrawal. It denotes a clearing of toxins from the body of the patient who is acutely intoxicated and/or dependent on substances of abuse.” In the context of software addiction, detoxification involves avoiding the use of the addictive software by finding alternatives, by implementing control procedures to aid avoidance. Unfortunately, detoxification oft leads to relapses due to the nature of treatment dealing with the short-term consequences and side effects, detoxification doesn’t offer any response to the breaking of habits and routines that are developed with an addiction and therefore ultimately leads to relapsing. Lipton *et al* (1982) suggested that detoxification should be the steppingstone to enrolling addicts onto long term treatments, and while detoxification is an important procedure, on its own it is ineffective.

Cold Turkey: The online journal Healthline defines going cold turkey as the action of spontaneously ceasing an addiction altogether out of your own volition. Cold turkey is infamous for being ineffective and dangerous; by stopping an addiction immediately it triggers a response known as a withdrawal symptom which is a negative consequence caused by the body or the mind due to the absence of a dependency. In the context of the problem, cold turkey quitting an addiction to the Internet or to video games can leave an individual with withdrawal symptoms such as they behave erratically, they are not focused, mood swings etc. Withdrawal symptoms are more severe than regular side effects because the response of the body or mind is akin to that of an overreaction, additionally going cold turkey oft does not last very long and most addicts immediately relapse. Albeit, for some individuals going cold turkey can be incredibly effective as it inspires in them self-restrain and promotes self-control and accountability, smoking is an addiction which sees the greatest number of people trying to go cold turkey. (Healthline Journal 2019).

Tapering Off: is the process of weaning off an addictive substance (or activity) by gradually cutting down the dosage over a period, gradually resulting in the complete ceasing of the addiction in a safe and effective manner. Tapering off is effective because unlike going cold turkey it gives the body or mind sufficient time to acclimate to a reduction in dosage whilst simultaneously ensuring there are little to no withdrawal symptoms. Tapering off works by reducing the tolerance level to the addiction as the dosage is reduced, it can also be used in conjunction with other treatments such as detoxification; the detoxification is the initial short-term treatment that causes the addicted user to start cutting down their dosages progressively. However, much like going cold turkey there is a large chance of the treatment being ineffective due to the possibility of relapsing due to the severity of the addiction.

My solution will employ detoxification followed by tapering off; this composite strategy will cause a gradual cessation of the addiction to desktop software in a safe and theoretically effective manner which aims to reduce the chances of relapsing and suffering from withdrawal symptoms, therefore my solution takes onboard the above-mentioned advice of employing detoxification in conjunction with another technique. I will implement this strategy by allowing the user to set a time allocation for how long they want to use an application for, once this allocation has been met the application will be rendered unusable as it will be instantly prevented from running until the application is closed, the reason why I think this is an effective strategy is because it offers some control to the user and simultaneously removes some, meaning that the user will not be allowed to over indulge their software addiction but they will not be completely prevented from using the application, which will eventually cause a decrease in tolerance levels, allowing for them to eventually altogether be free of the addiction.

## 2.6 Review of similar solutions:

There are several popular mobile applications that are designed to treat and manage software addictions:

Antisocial is a free android app that works to treat software addiction to specific applications by allowing the user to block them from running, and it tracks the user’s smartphone usage, once a usage threshold is met it forces the user to stop using their phone by displaying a popup which overlays the User interface when they use a high usage software application. The anti-social app is partially effective at solving the problem of software addiction as it tracks the user’s usage and provide suggestions on whether the user should block a certain app.

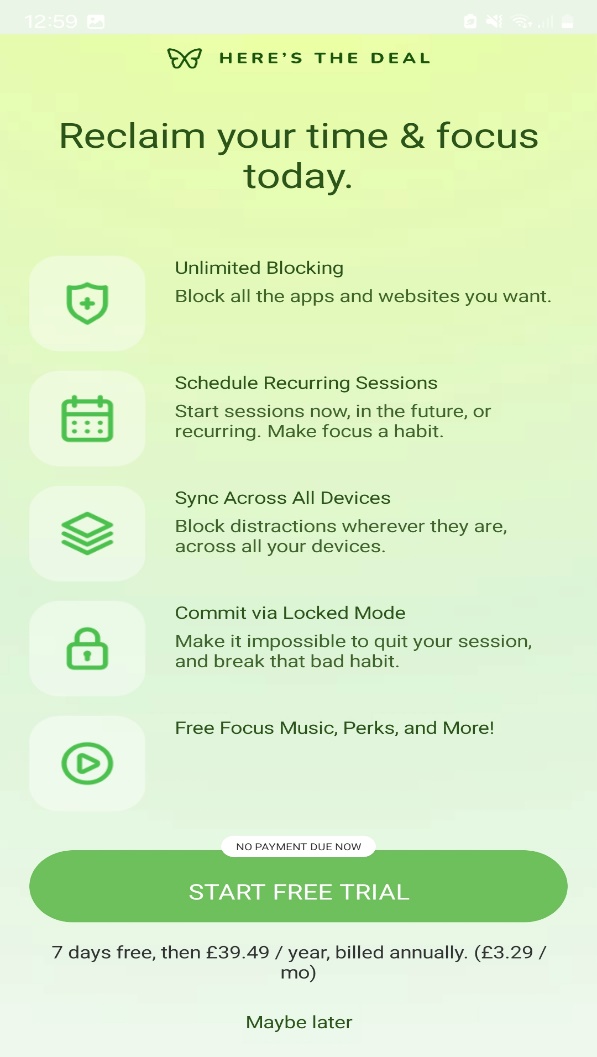
Off the Grid is another Android application which employs the strategy of completely blocking every application and function that the phone has between a user defined start and end time. Off the grid disables all notifications, blocks all phone calls, and texts and renders the phone unusable until the end time is reached and cannot be interrupted. This application provides a truly effective method to solve but at the cost of hindering the user, to explain my evaluation; the problem as the user cannot stop the application after starting it, however this does unnecessarily impact the user by rendering their phone useless, blocks phone calls and texts which can be urgent, and if they need to make a phone call they cannot; say in an emergency.

A screenshot of a cell phone

Description automatically generated

Figure : Off the Grid System Blocker

Freedom is a cross system application that works similarly to Antisocial where the user can create a list of apps that they wish to block from running, once created these apps are unable to be accessed until Freedom is uninstalled; freedom also has additional functionality of suspending the phone’s wireless functionality, which means the application simultaneously manages internet addiction. Additionally, freedom is not as intrusive as Off the Grid, it does not prevent receiving and sending texts and calls, however it is subscription-based application which means that once the subscription is over, if the user chooses not to renew, they will most likely relapse. Hence, the inaccessibility can cause relapses.



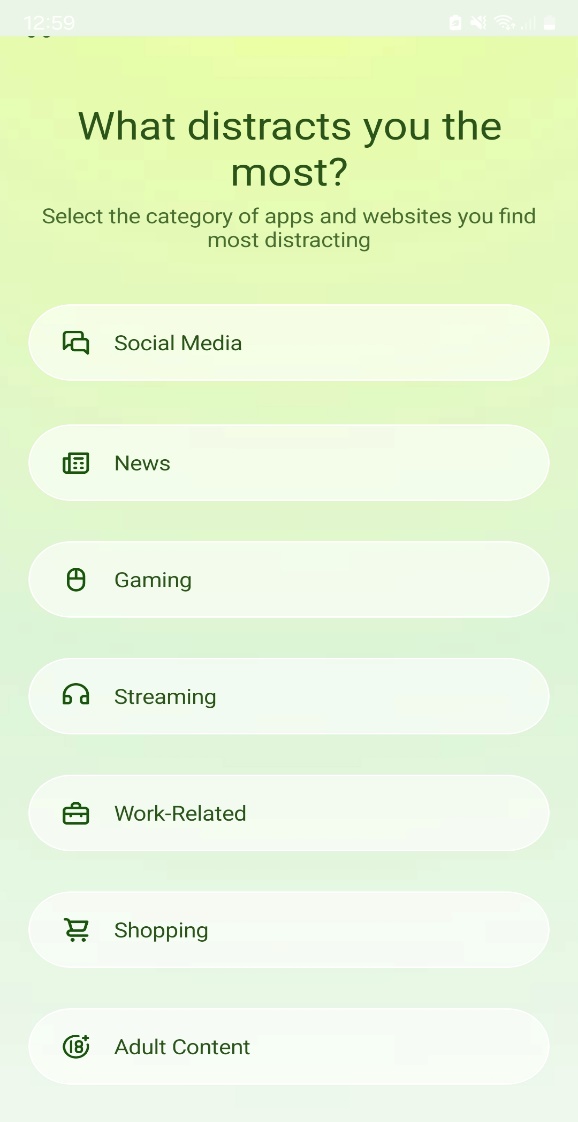


Figure : Freedom’s Customizability

Figure : Freedom Subscription Fee

The solutions above have their individual strengths and weaknesses, by combining all 3 solutions’ strengths and considering their weaknesses and correcting them, an effective software solution to desktop software addiction can be produced in the form of a GUI (Graphical user interface) application. The reason why the software practice of a GUI has been chosen over a CLI(Command-Line-Interface) is due to the development objectives requiring an intuitively designed, accessible and visually aesthetic interface which cannot be achieved with a CLI application.

## 2.7 Considerations:

Given the solution is an application which will be installed on the user’s computer, we have to comply with legal requirements, such as declaring the software as Open source so that users can install and use it without a license and ensure the software is secure from unauthorised and malicious actors, the application will not collect any data from the user. Ethical requirements need to be followed, such as ensuring the software is safe to use and does not have any adverse effects on the user’s computer.

# 3.0 Methodology:

### 3.0.1 Risk Assessment:



Table : Development Risk Assessment

Prior to starting development, a risk assessment was conducted to identify and analyse potential pitfalls that could possibly be encountered.

### 3.0.2 Design Specification:

Building upon the research that was conducted, the following design specification was constructed:

Functional: Implements behavioural modification techniques such as Tapering off and De-toxification to grant the user the ability to control their addiction over select software applications.

Aesthetics: A GUI composed of various widgets such as buttons, text entry boxes, which the user will interact with.

Environment: For desktop operating systems such as Windows, Mac, Linux.

Materials: A programming language and a GUI development library to build the requisite application and provide the required functionality.

Performance: The application should ideally be lightweight; hardware resource utilization should be small due to the scale of the application and because of the compatibility objective.

Target Audience: Ideally individuals that are suffering from a software addiction however all users can find a use for the application.

### 3.0.3 Data Flow Diagram:

A diagram of a data flow

Description automatically generated

Design diagram 1: Showcasing the flow of Data in the application in some critical functions.

## 3.1 Resource Analysis and Justification:

One potential resource to produce the application is Tkinter, a GUI Library that is included within the installation of python, meaning that no external installation of modules or packages is required. The advantages of using Python and Tkinter are it is included in the python installation and is lightweight, therefore there are less dependencies on the project and the finished application will have low storage and low performance requirements allowing for a range of system configurations to be compatible. Additionally, Tkinter is used in many open-source projects because it itself is open source and therefore there is no cost or licensing fee. However, the disadvantages are Simplicity; it is not possible to create advanced widgets and graphics with Tkinter which can lead to boring looking GUIs, and there are stability concerns stemming from Tkinter being re-written from its original source code. (Moore, 2018)

Next, we have Java FX which is a GUI Development library that comes with the Java installation. Clarke, Connors, and Bruno (2009) argue the biggest advantage in favour of using JavaFX is that unlike many GUI libraries JavaFX is cross compatible and cross operating system compliant. Due to the nature of Java being a cross system language, JavaFX inherits this property and therefore if the artefact is created for desktop, it can also be used on mobile with no alterations needed which allows the project to be available to another userbase, but for now the project is solely for desktop environments. JavaFX is also easy to use and works in many ways like Tkinter and therefore has a shallow learning curve whilst maintaining enough functionality to where it is still popular for creating industry applications.

But JavaFX has limited complexity; there are limits to the visual quality of the User interface which is a result of JavaFX being older than many GUI libraries and tools, also the JavaFX community is very small in comparison to Tkinter and QT which means there is less available guides and tutorials, there aren’t many forums to which users can ask and answer question regarding JavaFX. Furthermore, there is a somewhat sizeable overhead due to the Java runtime environment and virtual machine which everything is ran on and therefore the final application would be much larger in storage size and ram usage in comparison to other tools.

Qt is an open-source framework that uses C++ to create enterprise level GUI’s, it is widely used in industry because of its advanced tools for creating visually stunning applications for a range of platforms; including Desktop and Mobile. The benefits to QT are the ability to create an enterprise level interface which has the potential for multi-faceted functionality and usability, Qt unlike JavaFX and Tkinter can create modern-esque applications which are aesthetically pleasing through the importing of complex multimedia and allowing the creation of vector graphics and immensely customizable widgets. Additionally, QT has many packages and can create applications for a range of platforms and operating systems hence achieving one of the objectives of the project, furthermore QT has an online forum where users can ask questions and provide solutions to problems. (Summerfield, 2010)

However, Qt has a large performance and storage overhead resulting in the application being harder to use and run, affecting accessibility. There is a steep learning curve to using Qt which means it will be more difficult to start the development phase and that more time will have to allocated to the development and the testing procedure to ensure by the end of the project there is a functioning artefact which isn’t a guarantee, additionally Qt is also difficult to use with excessive functionality and is over-engineered to meet the current aims and objectives.

I have decided to use Tkinter because it is appropriate for the requirements, with low performance and storage overhead and possesses the functionality to create the solution.

### 3.1.1 Methodology:

The software development methodology chosen for this project is the Waterfall approach, the reason why it was chosen was because of the parameters of the project; there is a fixed deadline for the project, there is no commercial userbase whose requirements and feedback will need to be accounted for to guide the projects development, the application’s purpose and use cases are fixed in order to treat software addictions and will not be changed over the course of development. Furthermore, the stages of the project such as testing, development, planning are already known, therefore there is no need of multiple scrum cycles.

## 3.2 Back End Development

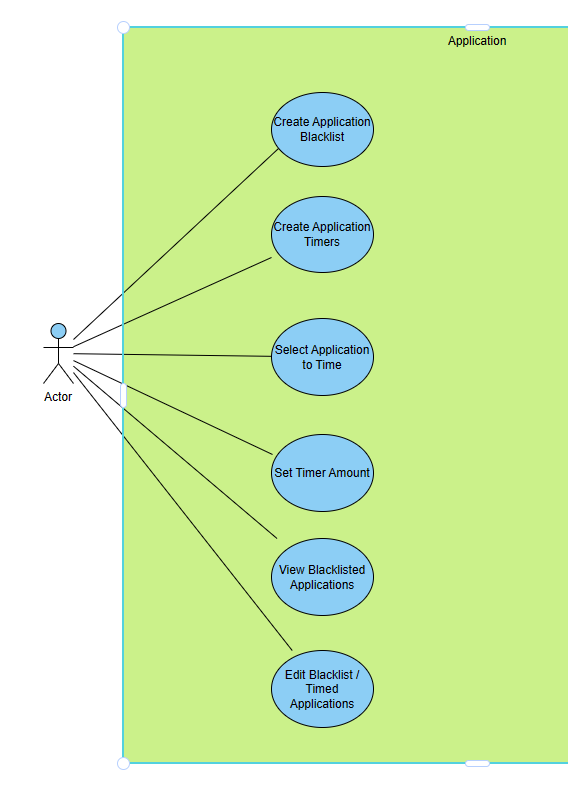


Figure 4: A Use case diagram depicting some of the application’s functionality, additional functionality will also be developed later.

The ‘Back End’ of an application refers to the area of the program, which is not directly accessed by the user, it is where the algorithms are defined and the logic is established. For the context of the problem, the back end will contain the solution to combat software addiction whereas the ‘Front End’ will be the area of the application the user interacts with. The figure above highlights some of the functionality that needs to be implemented, these functions will then be mapped to the front end of the application allowing the user to for example, select all the applications that they wish to time the usage for, set the timer for applications, Select the applications they want to prevent from running and being utilised as a way of going ‘Cold Turkey’ etc. The Back End was developed prior to the front end to iron out issues with the algorithm, to ensure that all back-end code was fully functional before being mapped to GUI Widgets.

### 3.2.1 Selecting the Application:

The user needs to be able to select which applications need to be timed, this functionality is facilitated by the File-menu of the operating system, the user can select the desired applications by navigating to the directory that holds the .exe (Executable) file, select a file by clicking it and pressing Ok, this then adds the application to an array-list data structure where it will be stored. Selecting the .exe file of the application via the File-menu is more robust and simpler than having the user write the name of the file in an entry box due to the chances of misspelling, or the executable having a name different to the application, therefore this approach is more user-accessible and helps to aid usability too.

csocsa
A computer screen shot of a program code

Description automatically generated

Figure 5: Displaying the Code for Open File Function which allows user to choose programs.

A screenshot of a computer

Description automatically generatedFigure 5 showcases that a ‘While loop’ is used so that the user can add several applications to the timing-list, which is an Array-List called “timeList”. Once the user is satisfied, they press the cancel button in the File-menu to exit the loop and save all their additions. The File-menu is accessed using the askopenfilename() function which belongs to the Tkinter library, we check that the file selected is an ‘exe’ by performing a Regular expression evaluation, if the evaluation returns true, we add the file to the timeList and display a message informing the user that said file has been added, if the file is another type we display another informative message to the user.

Figure 6: Screenshot of File-menu Selection

The use of screen popups is done to keep the user informed on the outcomes of each function and to promote understanding of what is happening and is implemented via creating messagebox objects and calling the function showinfo(). During the front-end development, this openfile function will be mapped to a GUI Button.

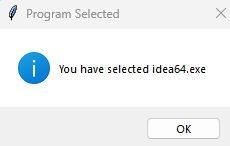


Figure 7: Popup of Success

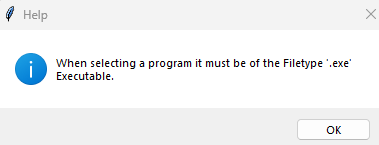


Figure 8: Popup when selecting non .exe

Figure 8 is displayed when the user attempts to select a non .exe file, the error is caught and the user is informed of the issue, which then returns to the File-menu.

### 3.2.2 Timing the Applications:

Once the user has exited the File-menu and saved their choice of applications, the next step is to enter a duration in minutes for the allotted time that they can use the application before it is stopped, to control their addiction in the form of automated tapering off and cessation. The reasoning for allowing the user to set their own limits is that it gives them control over their addiction when they previously had no control whatsoever, if the application were to set time limits for the user it would likely cause an immediate relapse and a subsequent deletion of the application from the user’s system.

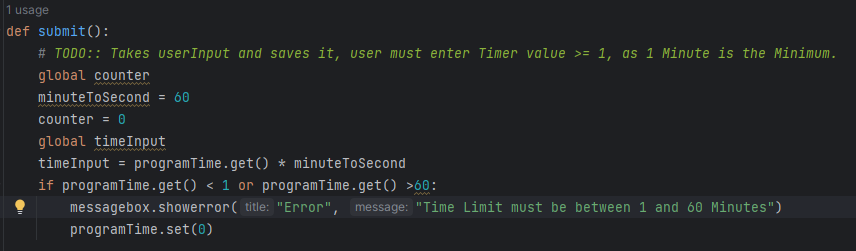


Figure 9: Code of the Timing Function

The Application asks the user to enter a time value which is between 1 and 60 minutes, any entered value outside of this range is rejected to ensure that the timer is set to a value that is not excessive and not viable; allowing for use beyond an hour will most likely result in the timer being ineffective as a method of treating addiction as there is no sensible limit and the user will be able to exploit this by setting a high timer value before the application is terminated hence there is no tapering off and no cessation of addiction. The user input is taken and multiplied by 60 to convert seconds to minutes. In the future the time limits may change depending on user feedback.

### 3.2.3 Terminating Applications:

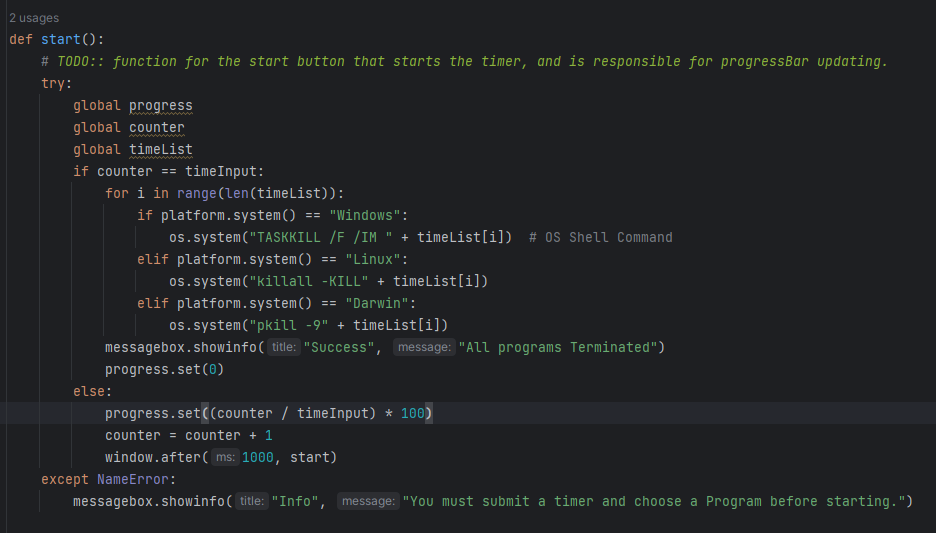


Figure 10: Timer Termination Code

The figure above showcases the code that triggers the applications termination, initially we check if the time limit has been reached, if so, we automatically terminate all the applications that are in the timing-list immediately. Once all Applications have been terminated, a popup is displayed on the screen. The applications are terminated by using an OS Shell command which terminates all the programs in the timing list sequentially. If the time limit has not been reached, the timer continues running and updating every second until the timer is met.

### 3.2.4 Blocking Applications:

Timing the applications to control usage is an implementation of tapering off; where limits are set to aid gradual cessation, the problem with solely using a tapering off approach is that in the cases of severe addiction it will lead to an immediate relapse which needs to be countered, and this is done via the implementation of detoxification through the blocking of applications from running after the timer ends.

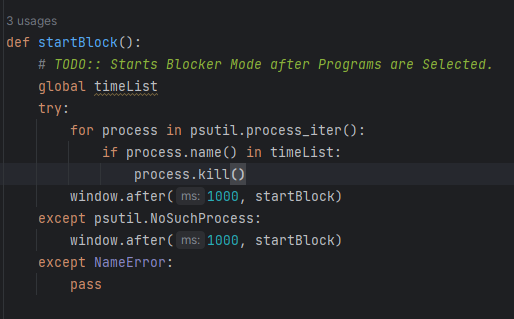


Figure 11: Code of the App Blocker

Figure 11 displays the blocking function’s code; it works by taking all the applications that are in the timing-list and keeps track of the state they are in; be it active or suspended or not running. If an application in the timing list enters the active state via being opened by the user after the timer, it will enter the operating system’s running processes list, which is when blocker executes an automated Kill task using the method process.kill() causing the application to automatically close and preventing the user from using it; hence detoxifying their addiction. This whole process is automated directly after the end of the timer and until the application is closed, effectively preventing relapsing by taking control away from the user via automation. Unfortunately, once the user has closed the application the blocker will no longer work, and this is a problem because the user can re-satisfy their addictive craving. To reduce the likelihood of the user closing the application the default close button has been disabled and the user must navigate to the options menu to exit, this is an additional obstacle to sway the user from closing the application.

## 3.3 Front End Development

### 3.3.1 Prototype Implementation:

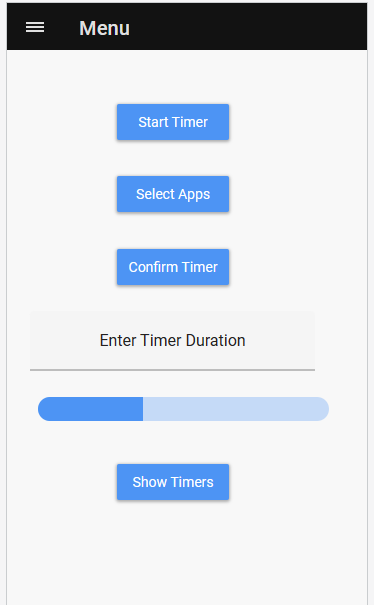


Figure 12: GUI Prototype

A screenshot of a program timer

Description automatically generated

Figure 13: Screenshot of the finished main GUI.

Figure 12 depicts the prototype of the GUI; Figure 13 showcases the finished user interface. The GUI is composed of buttons which are individually mapped to a back-end function, the GUI also has an entry field where the user will enter a value for the duration of the timer. The user entered value is then used by the Timer functions to time the usage of applications in the time-lists, below is a progress bar which represents the timer duration and progresses with the timer, eventually being filled when the timer has been met. The GUI could be more complex and aesthetically pleasing if QT was used, however the addiction treating functionality is still provided with a simple GUI.

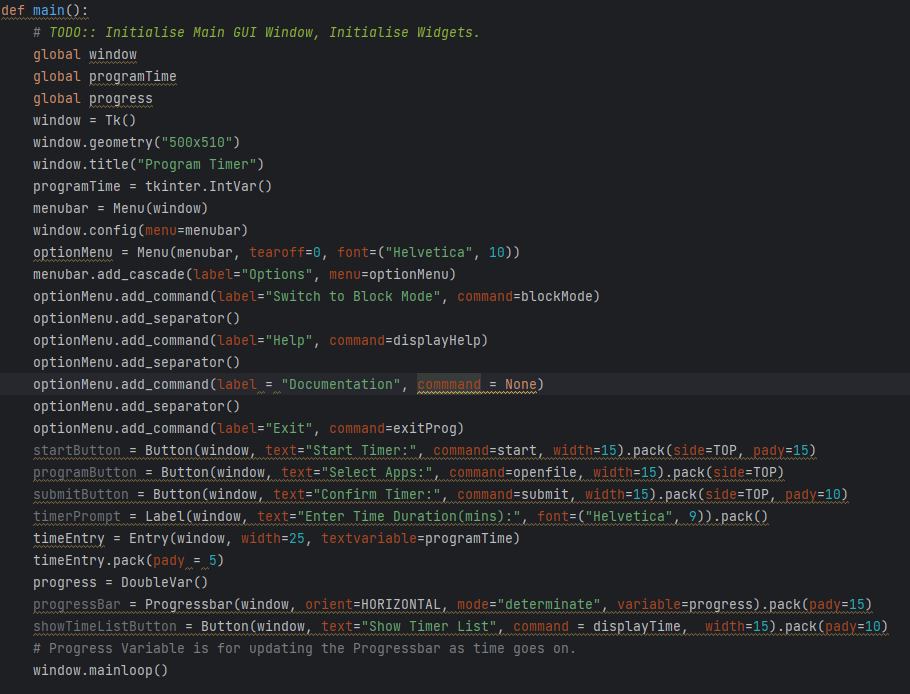
****

Figure 14: Code for GUI

Python is an Object-Oriented Programming language and as such all Tkinter widgets are objects; to create a button or any widget we simply instantiate an object and pass information such as the window object we are using, the widgets dimensions, the widget’s text, and we map a function to the widget which is actuated upon the user’s interaction. All widgets must have a window which is an OS rendered space for the application, acting as the canvas for everything else. To create a Button for example, we name the button button1 = Button () and a button will be added to the window, the same applies for other widgets. All Tkinter widgets can be mapped to an algorithm function allowing for easy functionality integration.

### 3.3.2 Blocker Mode:

In the case that the user simply does not need the timer and only the functionality to block specific applications from running altogether, the application has an option in the menu-bar titled “Blocker Mode”. Blocker mode is the second half of the solution which blocks applications from being re-opened after the timer ends, but now the blocker works instantly thus preventing the user from utilizing all applications that are on the Block-list until the application itself is closed, blocker mode works by creating a window separate from the timer to utilise the blocking functionality as part of a pre-emptive detoxification strategy.

A screenshot of a computer

Description automatically generated

Figure 15: The options-menu.

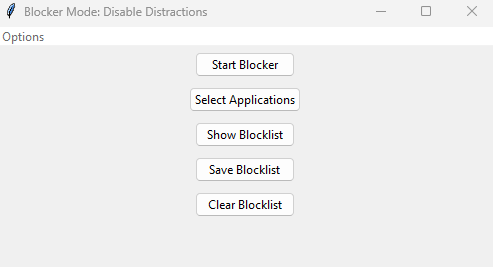


Figure 16: Blocker mode GUI.

Blocker Mode is solely composed of buttons which are mapped to specific functions, this design is due to focusing on simplicity, minimalism, and functionality. The options menu also contains a Help tab, a documentation tab, a return to the timer tab, and an exit button. To deter the user from closing the entire application the default close button has been disabled, and the user must close the application via the exit button in the menu-bar, this is a form of pre-emptive behavioural determent to stop the user from relapsing by making it less straight-forward to close the application.

### 3.3.3 Persistence:

One of the development objectives for the solution was for it to be persistent, persistency in computing terms is the process of the application saving and loading back the user’s configuration and changes to allow for easier utilisation by removing repetitive actions such as re-selecting the same applications to time and block after every exit of the application. Persistency greatly benefits the user experience and helps to improve reusability of the application. Persistency is achieved by writing the user’s configuration to a file, the configurations that are written are the Time-list and Block-list; the applications that the user has added to these lists is written to a file once they have clicked the “Save” button. Upon relaunching the application, the user will find that the timelist and blocklist are already configured, therefore if the user always wants to block the same applications, they do not have to go through the process of re-configuring everything every time and can just click save.

## 3.4 Challenges Faced:

During the development of the application blocker algorithm, challenges were encountered with keeping the application blocker active after the first iteration, the initial approach was to use a While Loop that had the condition of running forever which unfortunately resulted in the applications that were blocked to be re-active hence defeating the purpose of the blocker entirely.

A screenshot of a computer program

Description automatically generated

Figure 17: Code snippet of the While loop blocker

The solution to this problem was to use a programming construct called recursion which is like a loop, but it works without the needing of a loop condition and involves calling the blocker function from within itself and allows the blocker to execute continuously until the application is closed, and after every application is blocked from running the blocker automatically restarts hence preventing the user from relapsing. In figure 18, recursion can be evidenced by the line: Window2.after(1000, startBlock2)

This line of code implements recursion to keep the blocker function always active thus allowing it to detect if the blocked applications are running or not.

A screen shot of a computer program

Description automatically generated

Figure 18: Code snippet of the revised recursion app blocker.

Furthermore, another algorithmic challenge was the termination of applications after the timer had concluded, the initial attempt was to automate a sequence of key presses by programming a macro of “Alt + F4” to execute after the timer had ended, however this was insufficient as it did not terminate the applications that were being timed but rather terminated the Timer application itself. This made the application redundant, and it did not work on Linux or Mac. The solution was to use Shell commands which were executed by the OS to kill all the applications that were in the timer list, given the application was to be cross platform, the program would check the OS of the user and execute an OS specific command to terminate the applications in the Timer list via a loop.

Additional challenges involved the designing and implementation of the GUI; selecting a design methodology whilst also not exceeding my capabilities was cumbersome as one of the objectives of the project was to create an aesthetic user interface, and unfortunately due to the limitations imposed by Tkinter and by my skill level the objective was not fully achieved. However, a UI that was well designed albeit bland looking, which allowed the user to perform the necessary functions such as timing, blocking, and selecting applications to manage their addiction was produced.

Due to the default close button being disabled, there is a bug when re-opening the timer from blocker mode which prevents you from individually closing blocker mode, resulting in both windows being open simultaneously.

The greatest challenge encountered was trying to optimise the application for both Linux and Mac whilst only possessing a Windows machine, I was reliant on testing the application via virtual machines and focused developing the application on Windows, and then adjusting in the codebase to accommodate and port over the application for Linux and Mac. Unfortunately, this approach has resulted in certain bugs that are not apparent when using a Windows machine being apparent when using the application on Linux or Mac, hence this development objective was not fully achieved.

# 4.0 Testing and Evaluation:

## 4.1 Testing Methodology:

Regression testing, which is an on-going testing process throughout the development of the application, was carried out constantly before and after every increment of development to ensure the foundations of the application were all functional before the next component was developed. This process alleviated the troubles of testing and bug fixing as it was used in accordance with Unit testing every function to quickly resolve bugs and errors resulting in a functional application which the user interacts with, provides input to safely and correctly.

A white rectangular box with black text

Description automatically generated

Table : Displays a Small subset of the Test cases that were tested.

A screenshot of a computer

Description automatically generated

Figure 19: Showcasing Test case 2 Scenario of invalid entry.

A screenshot of a computer error

Description automatically generated

Figure 20: Showcasing Test case 1 where the user must enter a value between 1 and 60.

A screenshot of a computer

Description automatically generated

Figure 21: Showcasing Test Case 4.

A screenshot of a computer

Description automatically generated

Figure 22: Showcasing Test cases 3 and 4.

A screenshot of a computer

Description automatically generated

Figure 23: Showcasing the Result of Test case 4.

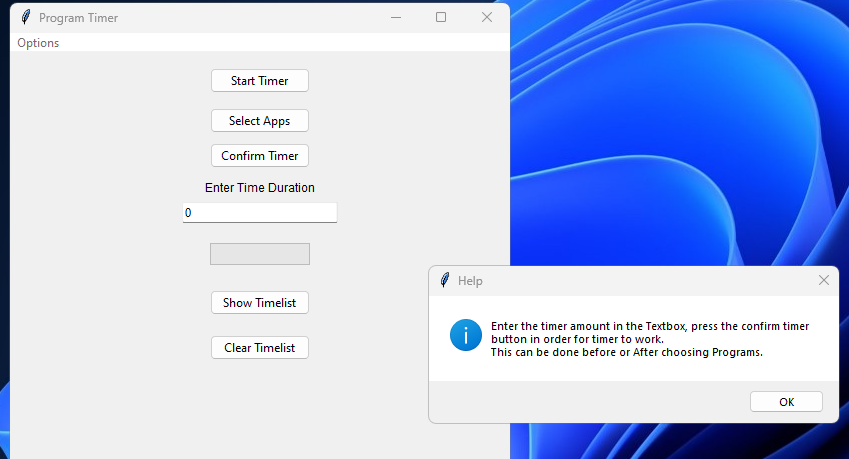


Figure 24: Result of Test case 7.

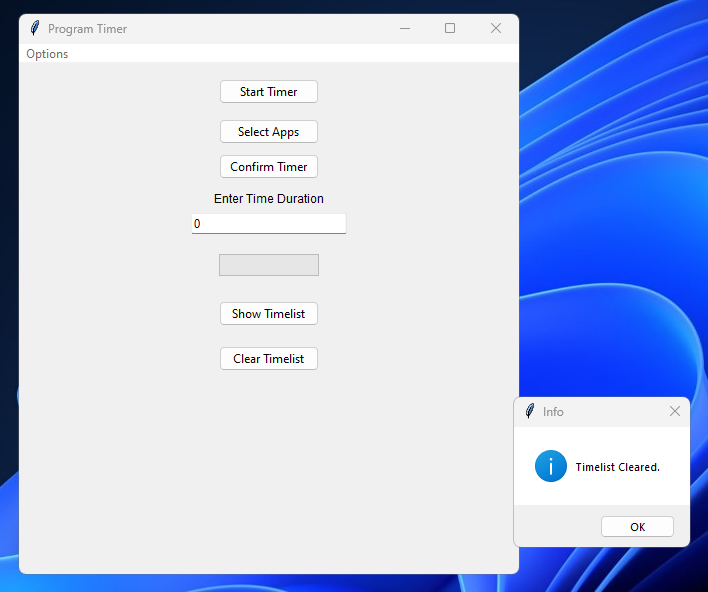


Figure 25: Test case 6, clearing Timelist.

The test cases in table 2 are a small subset of the total set of tests that were carried out, the tests in the Table are deemed to be the most important cases as they include the testing of both the front end and the back end functions simultaneously, therefore these tests extensively stress the integrity of the application as a bad result in any of those test cases would highlight the instability of the application as well as the poor quality of the behaviour modification algorithms implementation and application design. Furthermore, all the tests are carried out on the areas where the user will interact with the program. Overall, the testing process was streamlined as the test cases were established prior to the development phase of the application, the tests were designed with the objectives and the aim of the project in mind, also the constant regression testing after every iteration of development allowed for efficient development as most bugs and issues were discovered and fixed prior to continuing the development of the next components. The unit tests were carried out after every major increment of development to track the progress of development as well as providing an insight on what to focus on next.

## 4.2 Evaluation Methodology:

To see whether the solution developed adequately tackles the problem of desktop software addictions and achieves the project aim correctly and effectively whilst simultaneously achieving the objective of improving productivity, the application will be self-assessed to treat my addiction to Sekiro: Shadows Die Twice (A video game) over the course of the winter break. The application will be used to select the video game, the timer will be set for the maximum 60 minutes, and every day I will try my best to give up the addiction and actively avoid playing it. The project will be deemed successful if the following scenario occurs; I am able to give up playing the video game entirely and have gone at least a continuous 7 days without playing the game prior to the end of the break without relapsing. During the winter break I will record the date, the amount of time spent playing video games, the amount of time spent working on other commitments and the number of times the application was overridden for every day in a 3-week period. The application can be overridden by restarting the computer which admittedly is a design flaw hence why I will be keeping track of the number of computer restarts. The reasoning for why the evaluation period was during the winter break is because it allowed for the most accurate reading of how effective the application was at treating my addiction without outside influences, such as going to university, therefore this was chosen to get the most accurate reading of the application’s effectiveness.

## 4.3 Evaluation:

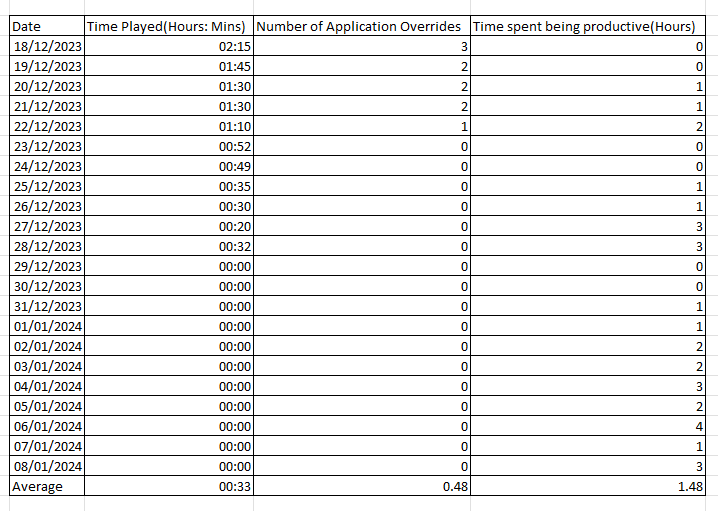
****

Table : Displaying a record kept over the 3 Week winter break of all the times video games were played, work was done, as well as the date.

Table 3 displays the date, the amount of time that I played Sekiro, keeps track of the number of times the application was overridden to allow me to play longer than allowed, and the last column details the amount of time that was spent productively completing coursework or studying. As can be seen the last day of winter break that I played Sekiro was on the 28/12/2023, and I played for a total time of 32 Minutes that day, and every day beyond I was able to completely give up playing Sekiro for a period of 11 days prior to the end of winter break. I have continued to uphold this feat of not relapsing, henceforth meaning that the evaluation objective was successfully met and that the application despite initially being in-effective, forced me to give up the addiction of playing video games, and one of the reasons why it worked was because it became a nuisance to override the blocker and the timer via restarting the computer entirely, compelling me to stick to the timer limits. Admittedly, the blocker was an excellent deterrent at preventing relapses whilst dealing with withdrawals, given that whenever I tried to re-open the timed applications, I was unable to and due to the removal of the exit button from the application, I was subsequently forced to detoxify from the addiction. As winter break progressed, I found myself witheringly unenthralled by Sekiro, I found myself spending more time completing coursework and studying for my Exams in the new year as well as spending an increased amount of time at the Gym and exercising, further proving that the application is a viable solution as it actively helped treat the effects of the addiction.

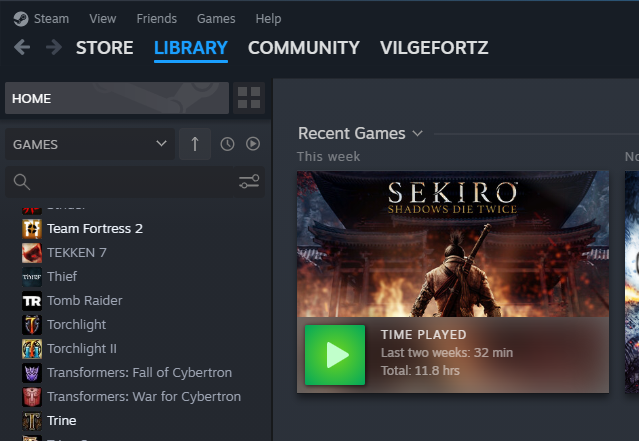


Figure 26: A screenshot showing Sekiro play time over the last 2 Weeks of Winter Break.

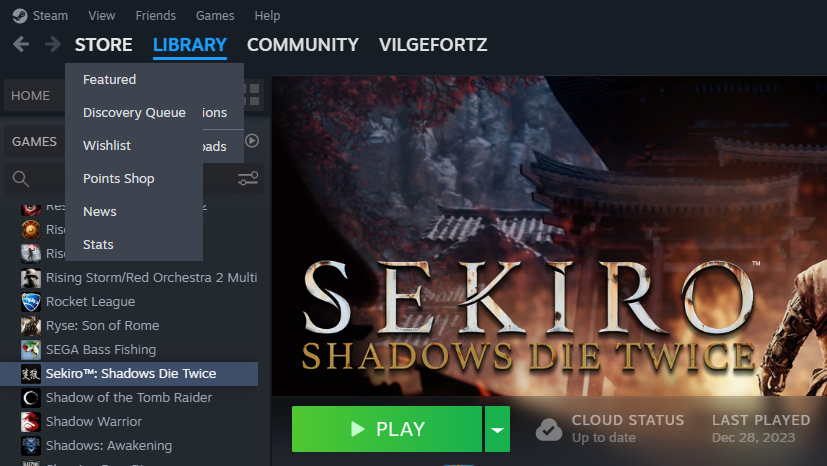


Figure 27: A screenshot displaying the date of last played for Sekiro December 28th, 2023.

Despite the evaluation objective and several development and research objectives being met, there were some that were partially achieved such as: The application’s GUI would be aesthetically pleasing, the application would work properly on all OS’s. The reason why these objectives were partially met are due to lacking both experience and the requisite skills to complete them. In the case of making the application seamlessly cross operating system, due to lacking experience with Linux and Mac OS development as well as lacking a Mac and Linux device, the application is more robust and optimised with less bugs on Windows, albeit the application is still functional on Linux and Mac the experience is less refined and contains more bugs. The limitations of the User interface are partially due to my inexperience with creating GUI applications as this project was my first endeavour in doing so, and partially due to the lack of tools and features with Tkinter, as discussed there is a limit to the complexity of what can be produced and the design control that is provided with Tkinter and this is because it is a small library and not a dedicated toolkit for creating large scale or enterprise level GUI applications.

Furthermore, in comparison to the other solutions reviewed earlier the application corrects their flaws in not combatting relapses by enabling both the timer and the blocker together to combat relapsing, which is why the solution works better as a long-term treatment for desktop software addiction, additionally the application gives the user less control over the application, preventing the user from constantly abusing the application and making it redundant. However, all 3 of the other solutions are cross compatible with Android and iOS and are much more visually appealing in terms of the UI.

# 5.0 Conclusion

To summarise, the aim and purpose of the project was to utilise software development principles and methodologies to create a software solution to treat and tackle addictions to software on the desktop via implementing behavioural modification techniques. The development objectives of the project were; the software produced would have an aesthetically pleasing GUI, the application would be intuitively designed to promote accessibility and usability, it would have minimal bugs and issues, it would be usable on Mac, Linux and Windows, the program would allow for persistency in the form of allowing the user to save configurations, documentation would be produced and integrated into the application as well as a tutorial to aid the user and there would be a menu system.

The methodology chosen was the waterfall methodology as it suited the project better, the behavioural modification techniques that were implemented are: gradual cessation, detoxification and tapering off, these techniques were implemented in the form of an application timer and blocker which can be used individually or in tandem with each other.

In conclusion, the application has achieved the principle aim of the project which was to produce software that can be used to treat and tackle desktop software application addictions, this was done by giving the user the tools to control their usage and combat withdrawals and relapses via behavioural modification techniques. Additionally, from the evaluation above it helped me overcome my own addiction to desktop video games as I was able to detoxify and gradually cease my play time over the winter break and eventually give it up completely and focus on beneficial activities such as going to the Gym. Moreover, many of the development and research objectives were achieved such as: An intuitively designed interface that promoted increased user accessibility and usability, a functional menu system was integrated into the solution which was used for easy navigation and for providing extra functionality, the application allowed for the user to save the list of apps that they were addicted to permanently into the time or block lists via the saving and loading of configuration files to inhibit persistency. However, certain objectives such as the application being cross platform and the GUI being modern and visually aesthetically pleasing were partially achieved due to limitations of ability and knowledge.

Overall, I have learnt extensively from the project especially regarding the complexity of addictions; why addictions are so difficult to treat, the endless consequences of different addictions, and I have developed an increased empathy to those who suffer from addictions as I now realise the extent at which it affects an individual. In terms of skills, I have greatly developed my prototyping and application design, testing and planning skills as well as others. I have also greatly stressed my programming ability and feel confident in the future creating more GUI applications.

## 5.1 Future Work:

The application is currently in a viable, usable state and can be used whilst experiencing minimal bugs on Windows, however I would like to completely overhaul and rebuild the application from the ground up in a different language to modernise and correct the flaws of the project like the poor resource utilization relative to the scale of the application, the lack of cross operating system optimization and tailored development, and the lack of an aesthetic GUI. I would decide to revisit my earlier choice of QtFrameowrk and C++ because of the extensive tools and functions available as well as C++ being a versatile language with dedicated libraries for tailoring development to specific OS’s. Additionally, C++ allows for dedicated memory management due to being a compiled language with the feature-set to do so, whereas Python is interpreted, and you cannot have control over memory allocation. Furthermore, I would like to include additional behavioural modification techniques to diversify the range of options for the user, therefore I would carry out additional research regarding addiction treatments. Furthermore, I would like to expand the reach of the application to the Mobile platform, as there are far more individuals addicted to their phone than there are addicted to the desktop, hence there is a greater need for the developed solution’s methods.

# 6.0 Project Management Review

To ensure that the project was finished by the due date various project management tools were used to allow for efficient and productive development, research, and planning. The first tool that I used to thoroughly identify and allocate potential tasks for the project as well as plan the predicted time for which these tasks will be started and completed by was a Gantt Chart.

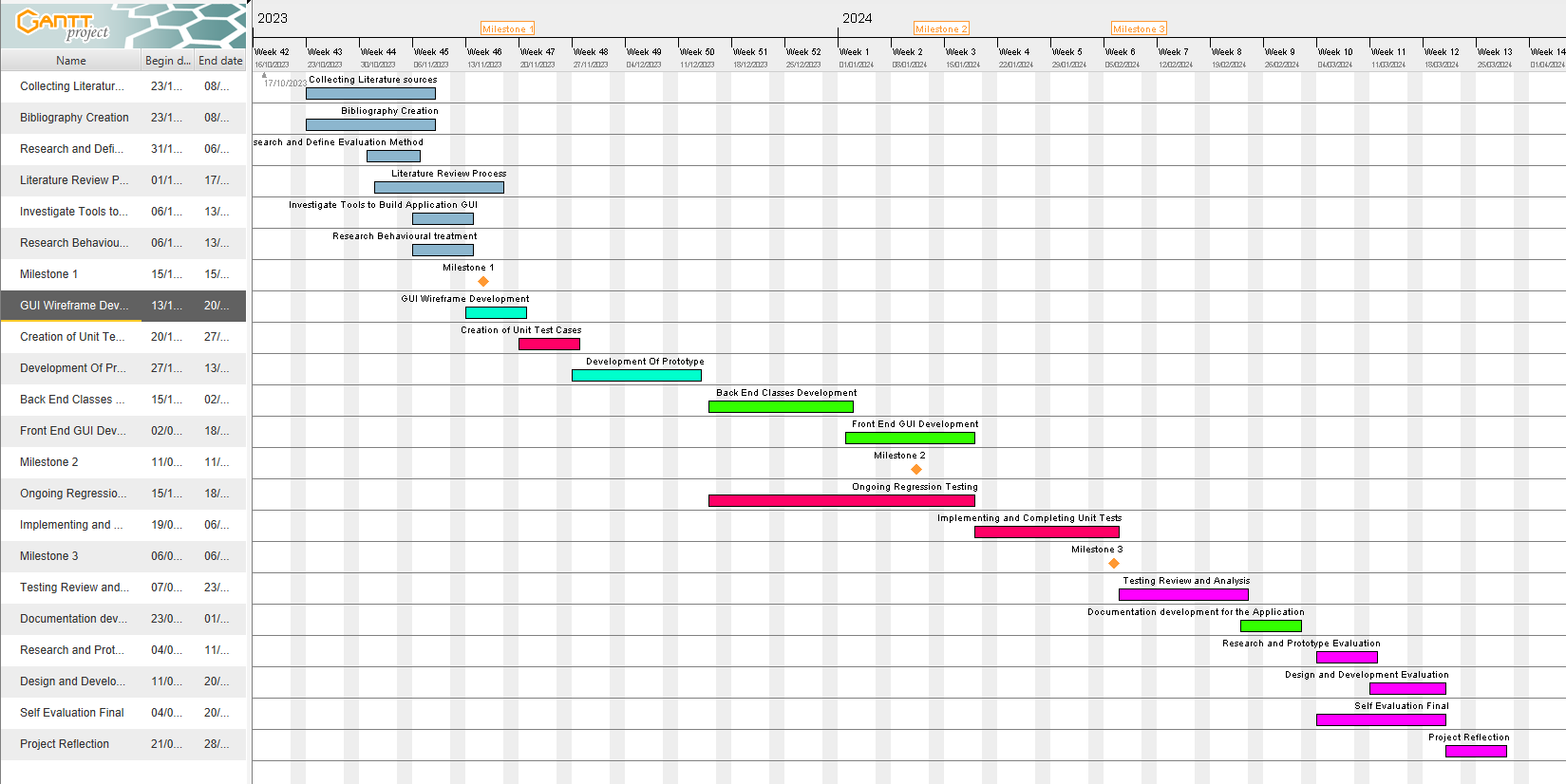


Figure 28: A Gantt chart that was created at the start of the project, showcasing the potential tasks.

Figure 28 conveys the early assumptions of tasks that were to be undertaken over the course of the project. In contrast with Figure 29 below which was created slightly later in the project’s lifecycle, it is brief and vague and does not give a detailed insight into the project, however it is to be considered as a basis for Figure 29 which is much more detailed regarding all the tasks that were completed over the course of the project. Figure 29 is also more accurate regarding the scheduling of tasks; as can be seen there are multiple occasions where different tasks overlap one another which is exactly how the project developed as I had to multi-task thoroughly to stay on track, to compensate for occasions of illnesses and other commitments. Furthermore, the dates at which each task commenced were strictly adhered to, so that the progress of the project was not impeded regardless of whether prior tasks were fully finished. Without the creation of the Gantt chart and without it being a detailed guideline of what to do and when, the project would’ve suffered greatly in both its quality and its completion.

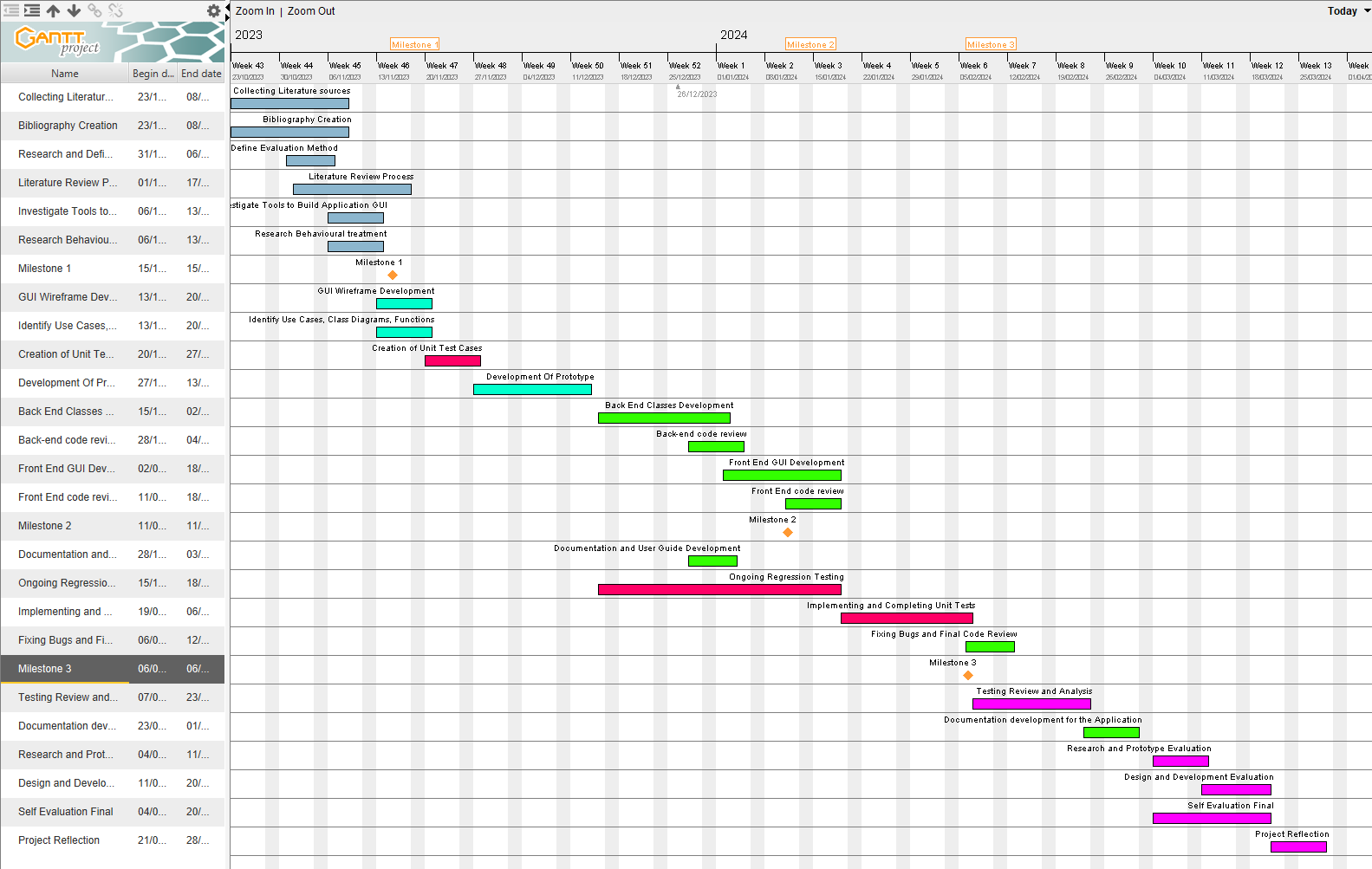


Figure 29: The revised Gantt chart which showcases ALL the tasks in the project to be done.

However, the effectiveness of the Gantt chart is largely dependent on the choice of Project methodologies; given that I had chosen the Waterfall methodology, it allowed for much easier identification of the project tasks and stages because development is not done in cycles but as a linear sequence.

With every iteration of the development phase, new bugs and issues arose and in certain cases the changes made rendered the project in-operable and ceased development. Fortunately, the version control and backing up of the project was done via GitHub, which allowed me to restore the last working version of the project, this allowed for easy development tracking. Without Github, the progress of the project would have been impeded due to additional time spent on bug fixing large changes as opposed to rolling back to prior working versions.

A screenshot of a computer

Description automatically generated

Figure 30: Showcasing the Github Repo for the project with Commits.

Overall, my time and task management with regards to the project was exceptional as I was able to balance other commitments alongside the project work with ease, this was largely due to putting in the effort to adhere to Gantt chart as well as devising a timetable to allocate when work will be completed, this allowed for swift development, testing and evaluation and contributed massively in finishing the project at the deadline. I made great use out of various project management tools, such as Github for Version control and back up, the Gantt chart for extensively planning out every stage. However, an area where I have noticeably lacked in was frequently communicating with the project supervisor, this was due to having to compromise and adjust because of both our schedules, I feel that next time I would like to make a greater effort in organizing and attending supervisor meetings to ensure that I can better receive feedback and help on the project.

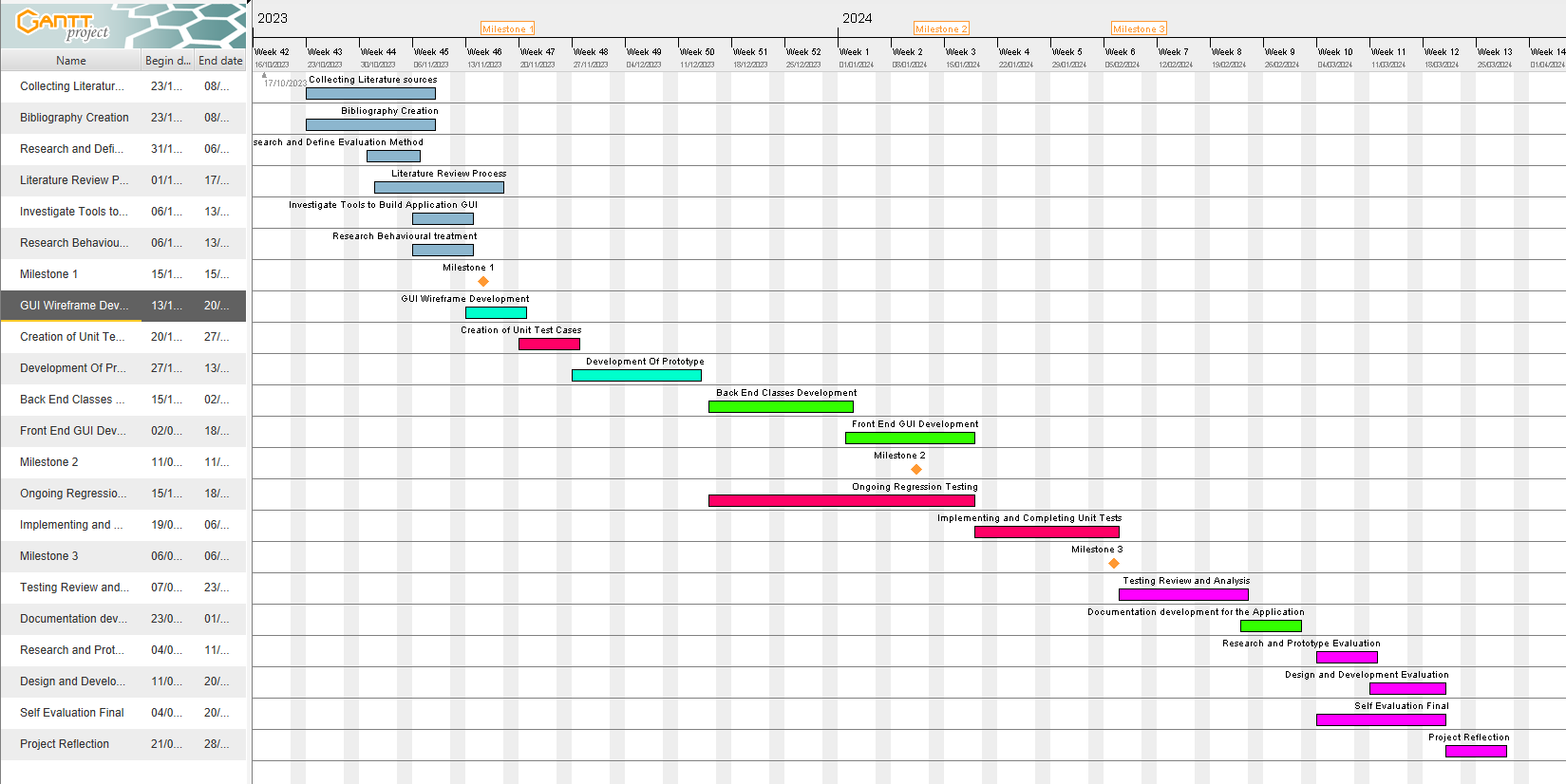
# References:

* Oxford University Press, 2020, Oxford Dictionary
* American Society of Addiction Medicine (2019). Definition of Addiction. [online] ASAM. Available at: <https://www.asam.org/quality-care/definition-of-addiction>.
* Wein, H. (2017). Biology of Addiction. [online] NIH News in Health. Available at: <https://newsinhealth.nih.gov/2015/10/biology-addiction>
* Heather, N. (2017). Rethinking addiction. [online] BPS. Available at: <https://www.bps.org.uk/psychologist/rethinking-addiction>.
* Vohs, K.D. and Baumeister, R.F. (2009). Addiction and free will. Addiction Research & Theory, 17(3), pp.231–235.
* Hoffman, R.S. and Goldfrank, L.R. (1990). The impact of drug abuse and addiction on society. Emergency Medicine Clinics of North America, [online], pp.467–480. Available at: <https://pubmed.ncbi.nlm.nih.gov/2201515/>.
* Butler Center for Research (2021). The Brain Disease Model of Addiction | Hazelden Betty Ford. [online] Hazeldenbettyford.org. Available at: <https://www.hazeldenbettyford.org/research-studies/addiction-research/brain-disease-model>.
* Heather, N., Best, D., Kawalek, A., Field, M., Lewis, M., Rotgers, F., Wiers, R.W. and Heim, D. (2018). Challenging the brain disease model of addiction: European launch of the addiction theory network. Addiction Research & Theory, 26(4), pp.249–255.
* Hall, W., Carter, A. and Forlini, C. (2015). The brain disease model of addiction: is it supported by the evidence and has it delivered on its promises? The Lancet Psychiatry, 2(1), pp.105–110.
* Griffiths, M. (1993). Fruit machine addiction in adolescence: A case study. Journal of Gambling Studies, 9(4), pp.387–399.
* Griffiths, M. (1995). Technological addictions. Clinical Psychology Forum, 76, 14-19. www.academia.edu. [online] Available at: <https://www.academia.edu/751805/Griffiths_M_D_1995_Technological_addictions_Clinical_Psychology_Forum_76_14_19>.
* Savci, M., & Aysan, F. (2017). Technological addictions and social connectedness: Predictor effect of Internet addiction, social media addiction, digital game addiction and smartphone addiction on social connectedness. Journal of Psychiatry and Neurological Sciences, 30(3), 202–216
* Sherer, J. and Levounis, P. (2022). Technological Addictions. Current Psychiatry Reports, 24(9), pp.399–406.
* Mondal, A. and Kumar, M. (2018). A study on Internet addiction and its relation to psychopathology and self-esteem among college students. Industrial Psychiatry Journal, [online] 27(1), p.61.
* Cai, J. and Tong, Q. (2022). Anatomy and Function of Ventral Tegmental Area Glutamate Neurons. Frontiers in Neural Circuits
* Tülübaş, T., Karakose, T. and Papadakis, S. (2023). A Holistic Investigation of the Relationship between Digital Addiction and Academic Achievement among Students. European Journal of Investigation in Health, Psychology and Education, [online] 13(10), pp.2006–2034. doi: https://doi.org/10.3390/ejihpe13100143.
* Mind (2019). About OCD. [online] Mind.org.uk. Available at: <https://www.mind.org.uk/information-support/types-of-mental-health-problems/obsessive-compulsive-disorder-ocd/about-ocd/>.
* Mahmoud, O.A.A., Hadad, S. and Sayed, T.A. (2022). The association between Internet addiction and sleep quality among Sohag University medical students. Middle East Current Psychiatry, 29(1). doi: https://doi.org/10.1186/s43045-022-00191-3.
* Center for Substance Abuse Treatment (2016). 1 Overview, Essential Concepts, and Definitions in Detoxification. [online] Nih.gov. Available at: <https://www.ncbi.nlm.nih.gov/books/NBK64119/>.
* Lipton, D.S., and Maranda, M.J. (1982). DETOXIFICATION FROM HEROIN DEPENDENCY: Advances in Alcohol & Substance Abuse, 2(1), pp.31–55. doi: https://doi.org/10.1300/j251v02n01\_03
* Healthline. (2019). Quitting Substances Cold Turkey: Safety, Risks, and More. [online] Available at: <https://www.healthline.com/health/opioid-withdrawal/quitting-cold-turkey>.
* Moore, A.D. (2018). Python GUI programming with Tkinter develop responsive and powerful GUI applications with Tkinter. Birmingham, UK: Packt Publishing.
* Clarke, J., Connors, J., and Bruno, E.J. (2009). JavaFX. Pearson Education
* Summerfield, M. (2010). Advanced Qt Programming. Pearson Education.

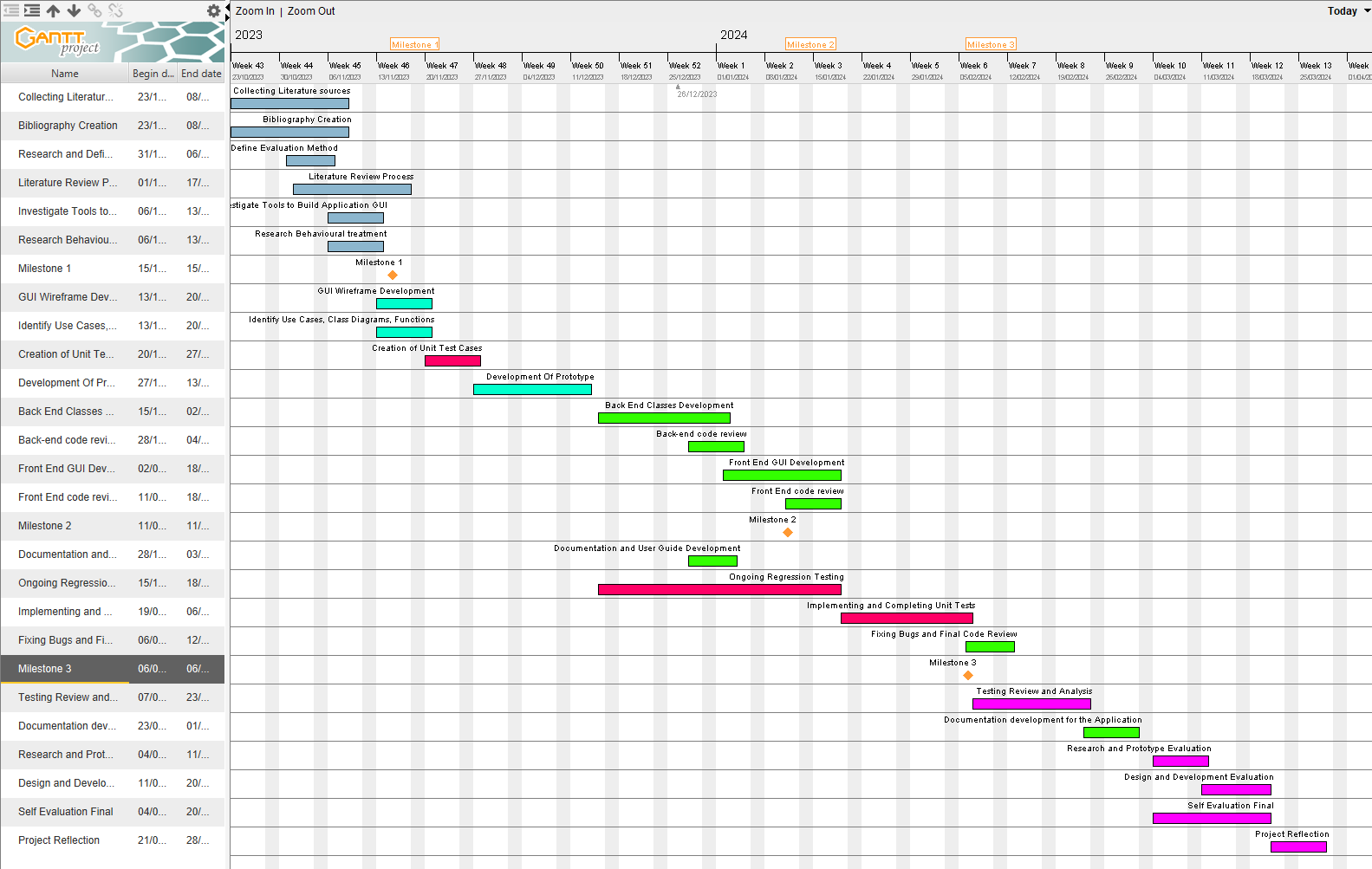
# Bibliography:

* All in References
* Koob, G.F. and Volkow, N.D. (2010). Neurocircuitry of Addiction. Neuropsychopharmacology, pp.217–238.
* Gerhart, N. (2017). Technology Addiction: How Social Network Sites Impact our Lives. Informing Science: The International Journal of an Emerging Transdiscipline, 20, pp.179–194.
* Bandawar, M., Narasimha, V.L. and Chand, P. (2018). Use of digital technology in addiction disorders. Indian journal of psychiatry, [online] 60(Suppl 4), pp. S534–S540.

## Appendix A: Initial Gantt Chart



## Appendix B: Revised Gantt Chart



## Appendix C: Testing Table

A white rectangular box with black text

Description automatically generated