Student Time Management App

ScheduleMate!

Student Name: Alfie Cooper

Student Number: 190082031

Module: CS3IP Individual Project

Tutor: Dr Joanna Lumsden

Aston University

24/04/2023

Declaration:

"I declare that I have personally prepared this assignment. The work is my own, carried out personally by me unless otherwise stated and has not been generated using paid for assessment writing services or Artificial Intelligence tools unless specified as a clearly stated approved component of the assessment brief. All sources of information, including quotations, are acknowledged by means of the appropriate citations and references. I declare that this work has not gained credit previously for another module at this or another University, save for permitted elements which formed part of an associated proposal linked directly to this submission.

"I understand that plagiarism, collusion, copying another student and commissioning (which for the avoidance of doubt includes the use of essay mills and other paid for assessment writing services, as well as unattributed use of work generated by Artificial Intelligence tools) are regarded as offences against the University's Assessment Regulations and may result in formal disciplinary proceedings.

"I understand that by submitting this assessment, I declare myself fit to be able to undertake the assessment and accept the outcome of the assessment as valid."

Contents

1 Introduction

A student’s education significantly impacts one's future by equipping individuals with essential knowledge, skills, and critical thinking abilities, which can lead to better career prospects and personal growth (Hanushek & Woessmann, 2006) . Educational institutions aim to enhance undergraduate student performance by focusing on developing academic skills often completed through teaching, feedback, and access to high-quality resources. However, non-academic factors, such as time management, are often overlooked, with studies indicating that at least half of all students face challenges in this area, including procrastination (Ozer et al., 2009, Solomon and Rothblum, 1984) with the prevalence of these issues continuing to increase (Steel, 2007).

Research has demonstrated that effective time management techniques can reduce procrastination and improve academic achievement (McKenzie & Gow, 2004) and (Trueman & Hartley, 1996). More recent studies however, emphasise the importance of teaching students to incorporate these techniques into their daily lives for maximum effectiveness (A. Kirillov et al. 2015, Adams R 2019). Consequently, this project aims to develop an application that utilises these techniques to assist students in their studies by enhancing their time management skills.

ScheduleMate is the proposed web application, which integrates multiple time management techniques derived from background research and knowledge elicitation methods. Developed using Node.js and Express frameworks, ScheduleMate features interconnected time management, scheduling, and goal-setting techniques. Additionally, the application analyses users' schedule and goal data, providing feedback on improvements through notifications. Furthermore, ScheduleMate incorporates a language model to assist users in finding solutions to time management issues they may encounter.

This report will outline the steps taken in this project, including background research and knowledge elicitation, to identify optimal time management techniques. The report will also cover the project's design, development, and testing phases. Lastly, an evaluation will be conducted to assess the project's success and the extent to which it meets its objectives.

2 Background research

Existing literature on the psychology of time management was reviewed to gauge the effectiveness of different time management techniques. Existing solutions were also reviewed to provide insight into how these techniques have been implemented in practical tools to support time management and how users have responded to them.

2.1 Literature review

2.1.1 Introduction

The primary objective of a time management app is to help users employ efficient time management strategies, allowing them to remain organised and focused on their objectives. In this project, it is essential to pinpoint the most effective time management techniques, specifically for students. A thorough literature review has been carried out to gain a deeper understanding of these techniques and their impact. The review explores previous research on various time management approaches and examines external factors that may affect their efficacy. Specifically, three time management techniques were investigated:

- Pomodoro Technique
- Time Scheduling
- Goal Setting

To expand the potential user base for the app, it was also important to explore whether different personality characteristics affect time management. Results from these studies could influence the app's functionality, increasing its usability for a wider audience. Therefore, towards the end of this chapter, studies into the relationship between characteristics such as personality; age and gender, are reviewed.

The final paragraph reflects the reviewed literature, where the results, limitations and recommendations for future research are discussed.

2.1.2 Method

The key resources/repositories searched were Google Scholar and Scopus along with IEEE (for more technical studies) and APA PsycArticles (for studies focusing on psychology). In addition, the Aston Library was searched and help from Aston Library staff was sought to improve the search methods used.

In selecting research papers for inclusion in the literature review, several criteria were followed to ensure the quality and relevance of the studies. Firstly it was crucial to consider only academic, peer-reviewed papers, as these have undergone rigorous scrutiny and validation, ensuring their reliability. Additionally, using studies with the most recent publication dates was necessary to avoid outdated information or methodologies that may have been superseded by newer research. Finally, the research papers considered were required to be based on empirical conclusions rather than anecdotal evidence to guarantee the robustness and validity of the findings.

When searching for research papers on time management techniques, keywords were utilised to ensure the most relevant and comprehensive results. The keywords used included more general terms such as Time management, Students, Pomodoro, Time Scheduling, Goal setting, SMART goals, personality, age, gender, and subject. By employing these keywords, the search focused on studies that examined various aspects of time management, such as specific techniques (Pomodoro, Time Scheduling, and Goal setting) and their effectiveness concerning students. Furthermore, the inclusion of keywords like personality, age, gender, and subject allowed for the identification of research that investigated how external factors might influence the success of these time management strategies. Overall, these keywords guided the search process, resulting in a robust and comprehensive collection of research papers that address the complexities of time management and its application in the context of students.

2.1.3 The Pomodoro Technique

The Pomodoro® Technique is a time management method developed by Francesco Cirillo in the late 1980s Cirillo (no date). It aims to enhance productivity by breaking work into short, focused intervals, called "Pomodoro's", followed by brief breaks. Cirillo (no date) defines several steps associated with this technique:

- Choose a task
- Set the Pomodoro® Timer to 25 minutes
- Work on the task until the Pomodoro® rings, then put a check on your sheet of paper
- Take a short break (Start with 5 minutes)
- Every 4 Pomodoro's, take a longer break."
- A default 25 minutes working length, which Cirillo suggests time can be set to anything.

Although Cirillo cites many benefits of this technique on his website, such as reducing interruptions and improving concentration, there is a lack of links to scientific evidence supporting his claims.

Further studies have been conducted to assess the effectiveness of Cirillo's methodology. One such study by Gobbo & Vaccari (2008) measured the effects of the Pomodoro technique in extreme programming teams. Whilst their results support the claims made by Cirillo, the authors fail to include much detail on the evaluation methods used to validate their claims. This study was only conducted on teams of software engineers. Although software engineers and students may face similar time management issues, students will typically work in different environments (study rooms, not offices) and have different motivations (securing a degree but no risk of getting fired). Students also tend to work more independently than teams of software engineers; it is, therefore, inappropriate to rely on this study as evidence that the Pomodoro technique would work for students in general.

Usman (2020) conducted an 8-month study which included regular interviews with students before and after implementing the Pomodoro method into their studies. Based on this study, Usman suggests, "*The PT® appeared to have addressed some of the reasons for multitasking, e.g. absence of break, pleasure and mood regulation, ease of multitasking. However, there were other reasons for multitasking that the PT® did not appear to have addressed, e.g. unconducive environment for study, absence of deadline/time pressure, peer expectations*" (2020, pg. 235). Usman (2020) also suggests that the factors governing the effectiveness of the Pomodoro technique are:

- Assessment deadlines

- Convergence of platform for the study
- Conducive environment for study
- Participants' active state
- Participants not adhering to the rules of the PT
- Allowing time to get used to the PT
- Group study

Usman only interviewed eight students, all of whom were studying scientific subjects. The small sample size and lack of variety mean his results are likely to exhibit bias, given that students from different subjects differ in their approaches to learning (Kember *et al.,* 2008). As such, Usman's findings only evidence that the Pomodoro technique addresses procrastination issues for (a few) students within a scientific field. For the results of his study to be more generalisable, Usman would have needed to increase his sample size and the variety of students interviewed.

A similar study with a larger sample size was conducted by Dizon *et al.* (2021), who tested 114 students before and after implementing the Pomodoro technique into their studies. Dizon *et al.* concluded that "*the application of the Pomodoro technique is more effective in reducing procrastination behaviour among nursing students*" (2021, pg. 63). A limitation of this study was that Dizon *et al.* (2021) used multiple time management techniques along with the Pomodoro technique, meaning that it cannot be clear if the Pomodoro was, in fact, the main contributing factor to their results.

2.1.4 Time scheduling

A study by (Bake *et al.,* 2018) investigates the effects of a scheduling intervention in an online class, focusing on its impact on student's academic performance, particularly those with poor time management skills. The intervention involved encouraging students to schedule when they would watch lecture videos. During the first week of the course, the intervention improved academic performance, with the positive effects being more pronounced among students who self-reported poor time management skills. However, these advantages faded in the following weeks of the course. It can therefore be argued that students follow their schedules less closely than in the first week.

While it is uncertain the reason for this drop-off, Bake speculates it could be because students' habits and other commitments became more established as the term progressed, making it more challenging to adhere to their planned schedules. Alternatively, it can be argued that students may lack the motivation to stick to a schedule for longer periods. The study also found that the positive effects of the scheduling intervention did not persist after its removal. This suggests that the intervention may have failed to successfully teach students lasting time management skills or induce a change in their time management behaviour.

In their 2018 study, Beattie and colleagues examined the factors differentiating between college students who thrive and those who struggle. The findings reveal that short study hours and cramming are the habits most closely linked to poor academic performance. According to the study, underachievers have difficulty maintaining their anticipated study hours as the term progresses. Text analysis indicates that these students are aware of their time management issues and express a desire to study more, yet fail to follow through. These findings suggest that students may achieve greater success by planning their time effectively. Moreover, the text analysis implies that unsuccessful students may lack the resources or awareness to identify and address their time

management problems and that an increased understanding of these issues could lead to improved academic performance.

Nonetheless, the study has limitations that could undermine its credibility. The majority of the findings rely on self-reported data, which may be subject to biases. Furthermore, the research sample exclusively comprised economics students, limiting the generalizability of the results to the wider student population. Another concern is the follow-up survey's lower response rate than the initial survey (83% vs 97%). This discrepancy could introduce non-response bias, as the characteristics or experiences of students who completed the follow-up survey may differ from those who did not. Beattie et al. (2018) attribute this difference to drop-outs, implying that the study's results do not account for failed students. This omission is significant, as it could be argued that these students will most likely struggle with time management issues.

Britton and Tessor's 1991 study suggests that students who demonstrate proficiency in short-range planning and time attitudes tend to achieve better academic outcomes. Interestingly, long-range planning, which involves setting goals for an entire quarter and organising work habits, did not correlate with the grade point average. Although this indicates that scheduling may positively impact short-term time management, it remains unclear whether planning or time attitudes drive this effect. Britton and Tessor's research has similar limitations to Beattie et al.'s 2018 study, including a sample consisting solely of college students enrolled in psychology courses, potentially reducing the generalizability of the findings to the broader college student population. Additionally, the time-management questionnaire relied on self-reporting, which can be subject to biases.

Another factor to consider is the study's publication date of 1991. Given that more than three decades have passed since its conduct, there may have been significant changes in the educational landscape, encompassing teaching methods and student characteristics. External factors beyond the educational context have also substantially influenced time management issues. For instance, as Liu F et al. (2022) highlighted, the proliferation of mobile phones has increased levels of procrastination, which could, in turn, affect students' ability to adhere to a schedule.

Wajcman (2018) suggests that dedication and review are required to make scheduling work properly. Wajcman study involved interviewing high-performance people such as Silicon Valley Engineers and Athletes who incorporate schedules within their daily lives. In roles where time management is essential, it can be argued that a lack of dedication and review of the student’s schedule may have been a reason for its lack of longevity. However, there are potential drawbacks to Wajcmans study. The lack of empirical data means the results may have been skewed due to unforeseen conditions such as subjectivity or researcher bias. Bake's results can be considered more accurate as they used quantitative data displaying a clearer picture of the effects of scheduling. Additionally, while one could view the implementation of time management techniques used by successful individuals as proven effective, it does not necessarily guarantee student results. As mentioned earlier, students may face different time management issues that are not prevalent in most professional work.

One study by B Tang (2020) involved him interviewing students one month and four months after implementing a time-blocking calendar into their studies. Tang states the results as “Four months after the implementation, a higher percentage of the student workers felt more positive about this scheduling system in that it contributed to:

- Improving their work attendance and efficiency.
- The perception of their life is more organised.

- The proficiency in using the online calendar tool which will benefit them in the long term.” - by B Tang (2020) Page 262

Contrary to Britton and Tessor's 1991 paper, this paper suggests that in the long term, using scheduling calendars positively affects students' time management. However, the lack of scientific evaluation methods indicates the possibility of inaccurate in this statement. As the results from this study were wholly based entirely on opinions, biases are not considered. To improve the accuracy of results, Tang should specify other characteristics of the students, e.g. demographic, age and subject, as these factors may affect the effectiveness of time schedules.

2.1.5 Goal Setting

Throughout the literature review on goal setting, it became apparent that Locke's goal-setting theory has significantly developed in this field and remains prevalent in most recent studies on goal setting. Locke (1968) proposed four ways of manipulating goals, which can be summarised as follows:

1. Instructions: Directly providing specific goals or objectives for individuals to achieve.
2. Time limits: Imposing deadlines or time constraints to encourage goal achievement.
3. Knowledge scores and competition: Indirectly influencing goals by providing feedback on performance and fostering competition among individuals.
4. Incentives: Using rewards such as money, praise, or participation to indirectly influence goal-setting behaviour.

In another study, Locke examined a decade's worth of laboratory and field studies on the effects of goal setting and performance. He found that 90% of the time, specific and challenging (but not excessively challenging) goals led to higher performance than easy goals. The results indicate that five goal-setting principles can improve the likelihood of success:

1. Clarity: Goals should be clear and specific, making it easier to understand what needs to be accomplished.
2. Challenge: Goals should be difficult but attainable to motivate individuals to put forth the effort.
3. Commitment: Individuals should be committed to their goals and believe in their ability to achieve them.
4. Feedback: Regular feedback on performance and progress can help individuals adjust their efforts and strategies.
5. Task complexity: Goals should consider the complexity of the task and allow enough time for completion.

A crucial point to consider from this research is that a significant contributing factor to the success of goal setting is how one defines their goal. While Locke (1968) also mentions external measures as influencing goals, this method might be impractical for a web application. As these findings are based on studies conducted in the 1960s, it could therefore be argued that they may not be as relevant in today's society, where technology has increased levels of procrastination among young people (Liu F et al., 2022), making it more challenging for students to adhere to their goals. However, Locke's (1968) theory is still applied in recent studies (Covington, 2000; Hambrick Hitt & Tucker, 2015), suggesting that these findings remain relevant.

Covington (2000) reviewed the literature on goal theory, examining how achievement goals influence school achievement through cognitive and self-regulation mechanisms. The research

demonstrated that students with learning goals tend to engage in more self-regulated learning, deep-level processing, and adaptive attributions for their successes and failures, leading to higher academic achievement. Importantly, while learning goals are positively associated with pride and satisfaction in success, they are negatively associated with anxiety in the event of failure. This supports Locke's emphasis on the importance of goal definition, as poor definition could reverse the intended effects. While this literature review provides a more recent overview of goal-setting research among students compared to Locke's (1968) paper, it is still 23 years old and may not account for recent changes within education or student characteristics.

Considering the importance of defining goals, it is essential to acknowledge any changes within the discussion of goal setting. A more recent preventive ideology on defining goals has been Rubin's (2002) 'SMART goals. Professor R Rubin states that for a goal to be clear and reachable, each one should be (Rubin R, 2002):

- **S**pecific
- **M**easurable
- **A**chievable
- **R**elevant
- **T**ime-bound

Rubin later proposed updating the acronym to 'SMARTER' by adding 'Evaluated' and 'Reviewed.' This refinement aligns more closely with Locke's definition, as he emphasised the significance of reviewing goals to enhance the likelihood of success (Locke, 1968).

Recent studies have assessed the effectiveness of SMART goals on students. A study by Bahrami et al. (2021) assigned tasks to two groups of university students, with one group having prior experience with the SMART goal program. The results revealed that "this brief intervention increased reported goal attainment" (Bahrami Z et al., 2021, p. 879). This study included a large number of participants of diverse ages and genders. However, a limitation of this research is that the effects of SMART goals were only monitored for a week. Consequently, this study cannot conclusively suggest that SMART goals are effective over more extended periods. As Wajcman (2018) highlighted earlier in this review, time management techniques can only be successful long-term if the individual remains committed.

Similarly, a study by Muñoz-Olano & Hurtado-Parrado (2017) examined the effects of SMART goal clarification on academic procrastination among college students. The results demonstrated that "participants who received the goal clarification treatment exhibited a significant reduction in academic procrastination compared to those in the control condition" (Muñoz-Olano & Hurtado-Parrado, 2017, p. 179). The significance of this study lies in its extended duration, in contrast to Bahrami's research, which provides further insight into the long-term effectiveness of SMART goals.

2.1.6 Personality Traits and Gender

Many of the studies previously examined provide limited information about their participants, which could be a significant oversight, as factors such as gender, personality, and age may influence the effectiveness of time management techniques. Consequently, this literature review has been expanded to assess research on factors beyond time management techniques that affect an individual's time management abilities. Exploring these additional factors is vital for better understanding user needs, enabling the development of a more inclusive and effective time management application.

A recent study by Albar et al. (2022) investigated the relationship between personality traits and learning strategies. Participants completed the NEO-FFI-3 questionnaire to identify dominant traits within their personalities, which included:

- Openness to experience
- Conscientiousness
- Neuroticism
- Extraversion
- Agreeableness

This study drew two conclusions. First, conscientious individuals exhibit better time management, while neurotic individuals would benefit most from time management due to their weaker focus and heightened anxiety levels (Albar R et al., 2022).

Secondly, the study revealed that females are more committed to learning and using academic resources than males. This may be explained by the reported correlation between females and the traits of agreeableness and extraversion (Albar R et al., 2022). Although Albar's study included a large sample size (309 participants), only medical students were involved. However, since the personality test results showed a wide variety among the group, the fact that all students studied the same subject became less relevant. Additional studies on this topic, such as Munt & Merydith (2012), were analysed for this literature review. However, these studies primarily reaffirm the findings of Albar et al., emphasising the consistency of the results across different research.

While this chapter examines certain factors that influence a student's time management, a more comprehensive overview of this topic requires further exploration of other factors. Additional studies should be conducted to investigate the impact of socioeconomic factors and other psychological aspects, such as motivation, self-efficacy, and stress. By broadening the scope of research, we can gain a deeper understanding of the complex interplay of factors that affect time management and develop more effective strategies tailored to individual needs.

2.1.8 Discussion

2.1.8.1 Pomodoro technique

While the Pomodoro Technique shows potential as a time management tool, the existing research has limitations that prevent a definitive endorsement of its effectiveness for students. Further investigation is necessary to determine the conditions under which the Pomodoro Technique may be most beneficial.

Nonetheless, the current research on the Pomodoro Technique can offer insights for developing requirements for a time management app targeting students. Usman (2020) indicates that factors like assessment deadlines, the conduciveness of the study environment, and group study situations can impact the effectiveness of the Pomodoro Technique. An app should enable users to customise the Pomodoro Technique settings (e.g., work and break intervals, notifications) to accommodate their unique study circumstances and preferences.

Dizon et al. (2021) discovered that combining the Pomodoro Technique with other time management methods yielded positive outcomes. A time management app could incorporate various complementary techniques, such as goal setting, task prioritisation, and progress tracking, to enhance the app's overall effectiveness. The Albar et al. (2022) study demonstrated that personality

traits and gender differences could influence time management and learning strategies. To address these differences, an app could integrate a personality assessment like the NEO-FFI-3, providing tailored recommendations based on individual traits and learning preferences.

2.1.8.2 Time scheduling:

The discussion of the various studies presented highlights the complexities of time management and the factors contributing to its effectiveness among students. The research findings, while insightful, have limitations such as self-reported data, subject specificity, and small sample sizes, which call for cautious interpretation. However, these studies can still provide valuable insights for the development of a time management app tailored to students' needs.

Bake et al. (2018) emphasise the significance of scheduling in enhancing academic performance, particularly for students with poor time management skills. The diminishing effects of the intervention over time suggest that the app should integrate features that sustain user engagement and motivation in the long term. For instance, the app could incorporate customisable reminders, progress tracking, and rewards systems to promote consistent use and adherence to schedules.

Additionally, Bake et al. (2018) imply that unsuccessful students might lack the resources or awareness to identify and address their time management issues, suggesting that increasing understanding of these problems could lead to improved academic performance. This notion aligns with Albar's assessment, which posits that males are less likely to adhere to a schedule as they tend to be less inclined to use external resources. As with the Pomodoro Technique, a time management application would benefit from customizability, allowing it to cater to a broader audience.

Wajcman (2018) and Tang (2020) stress the importance of dedication and review in ensuring the effectiveness of scheduling. Maintaining a well-suited schedule is crucial, as an overly simplistic schedule is inefficient. At the same time, one that is too difficult to follow could increase an individual's anxiety (Wajcman, 2018), negatively impacting time management (Albar et al., 2022). A scheduling app should incorporate features enabling students to periodically review their progress, assess their time management strategies, and adjust their schedules accordingly. This approach ensures that the app remains adaptive and relevant to the users' evolving needs, ultimately supporting the development of effective time management habits.

2.1.8.3 Goal setting

Research demonstrates that setting goals can lead to increased motivation towards achieving those goals. A key factor in this process is the accurate definition of goals. This literature review discussed various methods for defining goals, with the SMARTER goal framework emerging as the most relevant and evidence-based approach. As a result, it is essential for a time management application that incorporates goal setting to facilitate users in effortlessly defining their goals using the SMARTER criteria. Moreover, Covington (2000) identified that goal setting might exacerbate anxiety levels due to concerns about potential failure. Given that anxiety is linked to poor time management (Albar et al., 2022), it is crucial for a time management application to minimise this negative effect as much as possible.

To gain a more in-depth understanding of the impact of goal setting, further research is required. A significant area of exploration would be examining the long-term effects of goal-setting strategies. This information will enable the app to integrate techniques that promote sustained engagement, ensuring that users can consistently achieve their goals over time.

2.1.8.4 General

All these studies have limitations and potential biases, suggesting that they should be used cautiously. Many findings rely on self-reporting, which can contain conscious or unconscious bias and lead to inaccurate results. Another limitation is that the authors need to consider differences in demographics, age and economic status, which may impact the effectiveness of time management. Studies that do not consider different demographics may be unwittingly skewing data. However, Albar *et al.* (2022) provide an interesting argument for the lack of studies on personality in their research: “Employing personality sciences in the context of mentoring students is commonly faced with scepticism and concern that the risks of stereotyping outweigh the benefits of using them. Therefore, commonly many educators disregard personality and eliminate the use of its tools from their mentorship practice” (Albar *et al.,* 2022, page).

More studies need to be conducted to evaluate the long-term impact on students and better understand the effects of these time management techniques. These solutions are designed to be used throughout one's education. Therefore, knowing if a student can remain motivated to use these methods for extended periods is important. Additionally, more studies must analyse the relationship between time management techniques and different subject areas. Modern education offers a wide variety of different subjects students can study. Each subject has different types of study material and different methods of examination. Therefore, the time management technique's success will likely differ through different subjects.

This literature review has not evidenced a method of time management that is likely to be successful within different cohorts and for a long time frame. What it does do, however, is establish both pros and cons of certain methods. The next step in reviewing the current time management apps was using the evidence from this literature review to inform the design process.

2.2 Existing Solutions

A review of existing solutions for time management (mobile and web apps) was conducted to understand better how time management techniques have already been implemented into commercial products. Following the previous literature review results, it was important to analyse current applications that implemented all techniques already explored.

Only those with over one million downloads and a rating above four stars were considered to ensure the evaluation of high-quality applications. Examining popular and high-quality applications is essential, as it allows the assessment to better represent current market trends, consumer preferences, and features that strongly resonate with users. Furthermore, identifying key players enables a more effective determination of potential market gaps.

The search keywords employed included 'time management,' 'Pomodoro,' 'scheduling,' and 'goal setting.' The top three non-advertised, free apps that met the criteria were documented in a spreadsheet. Each application was then downloaded and tested individually. The subsequent chapter summarises each application and provides a detailed analysis of their respective features and design.

2.2.1 Forest – focus for productivity

URL - https://www.forestapp.cc/

Forest integrates the Pomodoro technique, enabling users to set customisable focus intervals (typically 25 minutes) followed by short breaks. Additionally, the app allows users to set custom focus durations and break intervals. Forest also provides a selection of ambient sounds, such as rain, forest sounds and cafe ambience.

Customisable focus intervals and breaks allow the application to cater to different work styles and preferences, aligning with Usman's (2020) study, which suggests that the Pomodoro technique is more effective when customised to fit the user's environment. Moreover, providing a selection of ambient sounds is advantageous for the app, as it helps users become less distracted by their surroundings. This is significant, considering Usman's (2020) study, which emphasises that a conducive environment for the study is essential for successfully implementing the Pomodoro technique.

Forest uses gamification throughout the application. If users maintain focus for the set interval, a tree grows; if they exit the app to use their phone for other purposes, the tree dies. Over time, users can grow a virtual forest, visually representing their productivity and focus. This feature encourages the user to re-engage and stay loyal to the app for extended periods. If a Pomodoro is not completed correctly, e.g. the user cancels the Pomodoro halfway through, a withered tree is planted in their forest. The aesthetically unpleasant 'dead trees' negatively affect the progress of the user's forest and are used to remind the user to complete Pomodoro's. However, this feature needs to consider situations where it is necessary to cancel a Pomodoro; for example, the user could have finished all their work for the day or needed to make an emergency phone call. This lack of flexibility may not resonate with users and discourage them from using the app.

The user receives coins when a Pomodoro is complete; the longer the Pomodoro, the larger the reward. Increasing the Pomodoro timeframe suggests that the user would get more work done; however, this might not be the case and could affect the student's efficiency. Parkinson's law suggests that work will expand to fill the time allotted for completion (Parkinson, 1955). Therefore this feature could harm students' efficiency as the increased timeframe does not guarantee a larger completion of the workload.

The app offers users detailed statistics on their focus sessions, including the number of trees planted, time spent focusing, and total forest size. Users can also view their progress over various time periods, such as daily, weekly, or monthly statistics.

This feature enables users to track and review their progress, which is crucial for effective time management, as demonstrated by studies (Rubin R 2002, Wajcman 2018). One advantage of this feature is that it facilitates goal setting, which has been shown to improve goal attainment (Bahrami Z et al., 2021). However, the goal-setting process within Forest lacks specific frameworks for defining goals. Covington's (2000) study highlights the importance of clearly defining goals, as vague goals can lead to performance anxiety for some users, negatively impacting their productivity. Although users can utilise the app's data to set goals using frameworks like SMART goals, Forest does not provide any built-in functionality to support this process directly.

The app enables users to tag each focus session with a specific task or category, which allows them to monitor the time spent on different activities. This feature promotes better time management and assists users in identifying areas where they may need to adjust time allocation. Nevertheless, a potential drawback of this feature is that categorising and tagging tasks could be time-consuming for some users, leading to decreased efficiency.

Forest's blacklist feature enables users to block specific apps or websites they find distracting during focus sessions. By adding apps or websites to the blacklist, users can prevent themselves from accessing these distractions while working, fostering a more focused and productive environment. However, according to user reviews, one issue with this feature is that the monitoring programs employed to track app usage during focus sessions can significantly drain battery life. This decreased battery performance may discourage users from utilising the app, as they might be concerned about their battery life during prolonged work sessions.

Users can collaborate with friends, join group focus sessions and share their achievements on social media, promoting a sense of community and accountability. This encourages users to stay focused and work together towards common goals. Users can plant trees together; if one fails, they both fail. Including accountability has the potential to motivate users to stick to their Pomodoro. However, a potential defect in this feature is the potential of peer pressure and competitiveness negatively impacting some users, causing stress and anxiety, negatively affecting time management (Covington, 2000).

A Chrome extension allows users to use the Pomodoro timer on their desktops. Being able to sync their mobile account to their desktop allows the user to add to their progress. A Chrome extension is a helpful feature that allows the user to progress even if they do not have access to or do not want to use their phone. However, the Chrome extension does not have full mobile app functionality.

2.2.2 Pomodoro timer – appfx

URL - https://play.google.com/store/apps/details?id=com.pomodrone.app&hl=en_GB&gl=US

The AppFX Pomodoro Timer application is a straightforward Pomodoro timer offering a minimalistic approach for users who prefer simplicity. It is simplicity, and limited features make it easy to use and navigate, containing only two pages: the timer and a settings menu. Users can set goals for the number of pomodoros completed daily, which may increase motivation. While the simplicity of this application makes it straightforward to use, from the literature review results, it can be suggested that its features are too simplistic to truly influence multiple aspects of time management (Usman, 2020). While it can be argued that the goal-setting feature debunks this claim, the application does not provide incentives or rewards for reaching set goals, which may limit motivation. Additionally, since users can set different lengths of Pomodoros, the number of completed sessions may not accurately reflect the amount of work done, rendering the goal-setting feature useless.

A large portion of the settings page is a grid with 20 different colour options for the UI, as shown in Figure 2.1, which takes up considerable space and offers limited practical value. Offering simpler options like light and dark modes would be more useful as it would increase space for other features.

The app will alert the user when the Pomodoro is completed by sounding an alarm and displaying a notification. This application provides a small amount of alert customizability; for example, a user can select whether the alert will vibrate. This function could be improved by making the alerts more customisable to match the user's environment. A user may not notice the notification within a loud environment, or it may cause the user embarrassment if the notification occurs within a quiet setting.

2.2.3 TimeTune Focus Planner

URL - https://timetune.app/

TimeTune is a time scheduling application designed to help users create and maintain customised daily routines to improve productivity and optimise their time management. One of TimeTune's key features is the ability to create reusable templates of time blocks. This is especially beneficial for students, as their schedules often consist of recurring events such as lectures, seminars and tutorials. Students save time and effort using templates, as they can reuse the same template for their weekly schedule rather than add each time block individually.

TimeTune's calendar synchronisation allows users to consolidate all their events in one place, providing a clear overview of their schedule. Students may use multiple scheduling platforms for their studies; for example, Aston uses the CELCAT calendar. Allowing students to display events from these calendars on their application easily improves organisation as all of their events are in one place; additionally, this reduces the time taken to set up the application.

However, as individual instances of the templates cannot be edited without affecting all the template instances, it is difficult for users to account for daily changes or interruptions. Users may, therefore, find the structured approach of TimeTune's scheduling system too rigid—especially those with more unpredictable schedules or who prefer a flexible approach to time management.

Another useful feature of TimeTune is the reminders and notifications system. This feature reduces the risk of a student missing an event. However, there is a potential downside to this feature. Frequent notifications could interrupt users' workflow and create distractions if not managed properly. However, TimeTune’s customisation feature minimises this issue and allows users to balance helpful reminders and unwanted interruptions.

2.2.4 Sectograph

URL - https://sectograph.com/

Sectograph is a time management application that offers several features to help users manage their daily schedule. Its defining feature, illustrated in Figure 2, is the clock-like circular design that resembles a clock face, making it easy for users to grasp their schedule at a glance quickly. This unique layout is also available for smartwatches, which is logical as the circular design uses the space on the watch very efficiently. However, on the mobile app, this design may not be a significant improvement over the standard schedule, as it only shows 12 hours of the day (24 hours is available for the paid version), and individual events may not be clear on the clock face due to the limited space.

Another potential issue with Sectograph is that users may need to read multiple tutorials to understand how the app functions once downloaded. Considering the app's limited functionality, forcing users to undergo these tutorials might be unnecessary and off-putting. Users who prefer more traditional calendar or list views or require more advanced scheduling features may not find Sectograph suitable for their needs.

Overall, Sectograph is a standard scheduling application that offers no more than TimeTunes focus planner other than its compatibility with smartwatches.

Figure 2.2 – Sectograph schedule

2.2.5 Dreamfora

URL - https://dreamfora.com/

Dreamfora is a goal-setting app that intends users to build sustainable habits to reach goals. For each goal, the user describes the habits the user needs to accomplish this goal and the tasks required to complete these habits. Additionally, users can add notes for each task and habit. Encouraging the user to break goals down and comment promotes the user to define their goal in more detail which is important for goal attainment (Covington, 2000). This feature additionally allows the user to track and reflect on their current goal progress. As mentioned in the literature review, feedback and reflection can improve the chances of users reaching their goals Locke (1968).

Dreamfora also includes features to help keep the user on track, such as reminders to complete tasks and motivational messages when the app is opened. Including such notifications in a time management application would be helpful as it reduces the risk of users forgetting to complete a task. Additionally, it can possibly reinforce a positive mindset that can reduce negative emotions such as anxiety and stress, which can hinder successful time management (Covington, 2000). However, constant pop-ups can become disruptive.

Dreamfora also offers premade goals where individuals can use templates instead of inputting tasks. This feature may increase ease of use; however, the tasks within the templates cannot be changed. As the tasks are not customisable, it is more complex for the user to reflect on their progress as they cannot make any changes. By making the templates customisable, users could change them based on what suits them, increasing the possibility of reuse.

One standout feature of dreamfora includes the ability to post their accomplishments to an online social media feed. This feature has the potential to motivate users, as once posted, users can receive recognition from others. However, the app also shows the feed of others individuals. This can lead to other time management issues, such as distraction and procrastination.

2.2.6 Loop habit tracker

Loop Habit Tracker offers several features to help users build and maintain good habits. Users can create habits and set how often they should complete these. Goal completion can be quantified as "yes or no" or measurable. Measurable habits allow the user to select more specific habits. For example, instead of "go for a run", the user could record the distance/time of each run; this is illustrated in Figure 3. This feature can therefore accommodate various types of habits.

This app is straightforward and has a clear user interface, as shown in Figure 2.3, making it very easy to use; however, it lacks functionality. This app helps record user habits; however, there are no functions to promote re-engagement or motivate the user to complete these habits**.** Therefore, users may struggle to maintain this app for more extended periods.

The application allows users to review their habit traits, by providing visual progress representation, streaks and habit scores; users, therefore, have more information on their progress to analyse. However, due to the lack of re-engagement features, the results may be inaccurate due to users not updating their progress.

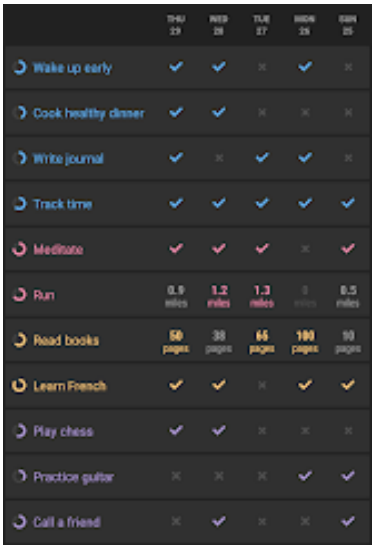


Figure 2.3 – Loop habit tracker main page

2.2.7 Trello

Trello is a web-based project management tool used to organise and prioritise tasks. The tool allows users to create boards for different projects and add cards for individual tasks. Although Trello mainly operates as a Kanban board, tasks can be visualised in various formats, including a calendar and weekly schedule. This makes the application acceptable to students who prefer different organisational formats. Figure 2.3 illustrates Trello's calendar display.

Trello makes it easy for individuals to edit their dashboards by implementing drag and drop. This makes it a lot easier for users to make changes to their dashboard compared to forms which other apps use. This flexibility will better account for sudden changes within one's schedule.

Trello allows users to categorise and filter tasks using colour coordination, allowing them to better visualise their tasks in terms of different categories. This can be useful within an app catering to students as tasks can be categorised into aspects such as subjects or assessments.

Trello enables multiple users to work on a project simultaneously, making it an ideal tool for teams to collaborate on tasks and projects. This feature is unnecessary for a student time management app as most of the work would be individual, it would be possible to include a collaborative feature similar to Trello's; however, this would rely on the other students using the same application.

Trello integrates with various tools and apps like Google Drive and Slack. This allows the student to customise their platform to suit their needs better.

Trello is also available on mobile, making the user’s information accessible away from their desktop. This would be useful for students as they can check and record progress away from their desktop, which is very suitable for more practical subject areas.

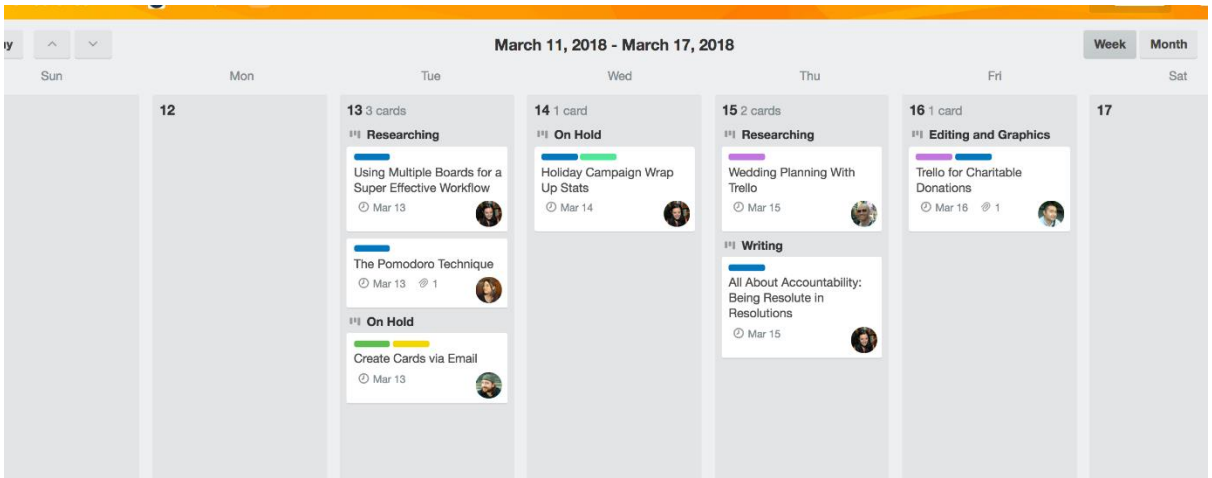


Figure 2.4 – Trello’s calendar display

2.2.8 Product Comparison Grid

A comprehensive product comparison grid

was utilised to effectively highlight the differences in features among the various time management applications, as depicted in Figure 2.5. The grid is organised with headers that categorise each criterion, providing a clear and systematic structure. The results from this comparison grid were utilised in our discussion to analyse and determine the most optimal ways of implementing time management techniques in various applications.

App	Forest	Pomodoro appFx	TimeTune	Sectograph	.oop habit tracke	Dreamfora	Trello
Average rating	4.75	4.8	4.35	4.5	4.7	4.5	4.3
Downloads - google play	10M	1M	5M	5M	5M	1M	10M
Pomodoro							
Custom focus time	X	X	X				
Multiple break length		X					
Completion notification	X	X	X				
Focus time analysis	X						
Completion reward	X						
Schedule							
Pre-made templates			X	X			
Add time blocks			X	X			X
Calendar sync			X	X			X
Notification for current time block			X	X			
Goals							
Study time goal	X	X	X			X	
Pre-made goal plans					X	X	
Goal reminder	X				X	X	
To do							
Categorize tasks					X	X	X
Prioritize tasks					X		X
Share tasks with others							X
Anti-distraction							
App restriction	X						
Website restriction							
Allow list	X						
Gamification							
Score	X					X	
Visualization	X			X			
Motivational measures							
Displays user progress	X	X					
Motivational messages	X					X	
Multi-user engagement							
Accountability	X						
Sharing progress	X					X	
Share tasks							X
Platform							
Mobile	X	X	X	X	X	X	X
Web App							X
Browser extension	X						
Other							
Custom notifications			X		X	X	
Analysis	X	X	X			X	

Figure 2.5 – Product comparison grid

2.2.8 Discussion

This discussion will compare the adoptions of time management techniques used by the applications and identify the best practices for a time management application for students.

2.2.8.1 Pomodoro

Time management applications utilising the Pomodoro technique significantly benefit from incorporating anti-distraction measures, as they help eliminate distractions during the timed work sessions. The majority of these measures involve blocking specific apps or notifications. This is particularly important for students, who may be more susceptible to distractions from their mobile phones (Liu F et al. 2022). Consequently, the Pomodoro technique is more effective in mobile applications, as they require administrative permissions to block notifications and other apps.

Pomodoro applications necessitate customisable notifications to inform users when a Pomodoro session or break is complete, allowing for better adaptation to the user's environment. These applications have implemented various features to enhance motivation, such as goal setting, motivational messages, and analytics displays. Customisable notifications cater to a diverse range of students who work in different study environments.

Regarding Pomodoro timers and goals, AppFx's Pomodoro timer demonstrates that it is more effective to quantify goals based on the amount of time spent studying rather than the number of Pomodoro sessions completed. This is because Pomodoro timers can be set to varying lengths, and completing more sessions may not necessarily result in a greater amount of work being accomplished within a given day. Furthermore, it is important to note that the number of study hours completed in a day is not equivalent to the amount of work done by the student, as it does not take into account work done outside of study sessions, such as attending lectures and tutorials, which are essential for students.

2.2.8.2 Time Schedule

Apps that include a schedule should have a clear design allowing the user to see all events clearly as opposed to Sectographs display. While making the application compatible with smart watches has the potential of improving accessibility to students' smartwatches. This market is too niche and it is better that the app is accessible without technological requirements.

Previous schedule issues include the risk of assigning too much or too little work. Schedules should be easily customisable, allowing users to find and assign the correct workload. Trello's calendars emulate this by incorporating a drag-and-drop technique. Additionally, students would benefit by having repeatable events similar to TimeTunes focus planner. This would save time when emulating weekly obligations such as lectures and seminars. Another feature of TimeTune is the ability to merge the application’s calendar with other calendars. For example, students may have meetings online using software such as Skype or Microsoft Teams. This feature would remove the need for students to add these events themselves.

TimeTune’s schedule involved goal setting, It can be argued that this may harm an individual's ability to stick to schedules as it could push them to add too much (Wajcman, 2018).

2.2.8.3 Goal Setting

Goal setting is a prevalent feature in the time management apps reviewed and is often linked with to-do lists. Encouraging users to break down their goals and add comments promotes a more detailed definition of their goals, which is crucial for goal attainment (Covington, 2000). This feature also enables users to track and reflect on their current goal progress. As mentioned in the literature review, feedback and reflection can significantly improve the chances of users reaching their goals (Locke, 1968). By incorporating these elements into time management applications, developers can create more effective tools that support users in achieving their objectives and enhancing their overall productivity.

Goal-setting applications provided a framework for the user to define their goal, which is important for its success (Covington, 2000). Examples of how existing solutions have implemented SMART are as follows:

- **S**pecific **–** Including a details input allows the user to add much detail to their goal description
- **M**easurable – Most goal-setting applications allow the progress of goals to be easily measured. The application 'Loop habit tracker' allows the user to specify the actual units, increasing the measurement's accuracy.
- **A**chievable – The application Dreamfora allows users to break goals into subtasks. Breaking down a goal into smaller tasks makes it clearer and more manageable, allowing the user to focus on one specific task at a time rather than feeling overwhelmed by the larger goal.
- **R**elevant – The user determines the goal's relevance; therefore, it would be difficult to include a feature that influences this measure.
- **T**ime-constrained – Dreamfora allows the user to specify the due date of each goal.

3Knowledge Elicitation

Although the prior background research established a strong foundation for understanding various time management techniques, the literature review and app analysis revealed gaps in existing knowledge and tools. The objective of this chapter is to utilize knowledge elicitation methods to gain a deeper comprehension of user requirements, as well as to identify the practices and tools currently employed by students and the impact they have on their time management. This will ultimately allow for the application's design and specifications refinement, ensuring a more effective and tailored solution for users.

3.1 Method

The questionnaire was selected as the suitable knowledge elicitation method for several reasons. Firstly, distributing questionnaires to a diverse group of students enables developers to collect a wide range of perspectives and experiences related to time management, enhancing the generalizability of the results. Secondly, the limited time frame for this project necessitated a rapid method for obtaining quality feedback, making questionnaires an ideal choice due to their efficiency and convenience.

Once the knowledge elicitation method was selected, the questionnaire had to be designed. The questionnaire was divided into four sections. Below displays each section along with the reasoning and objective of the questions;

Section 1 – Demographics

Questions 1-3 were used to record participants' age, subject of study and education experience. Collecting demographic information helps ensure that the sample is representative of the target population, increasing the generalizability and validity of the results. Additionally, demographic variables can reveal patterns and trends in time management behaviours, strategies and tool preferences across different groups. By understanding the preferences and needs of students across different demographics, an application can be developed to offer content tailored to the specific requirements of each discipline, resulting in a more effective and user-friendly experience.

Section 2 – Techniques and Challenges of the Student's time management

Questions 4-5 inquired about participants' typical time management practices concerning their academic studies and their challenges in managing their time for these studies. Employing open-ended questions for these topics enables participants to offer more comprehensive and in-depth responses, revealing subtle aspects and intricacies in their time management habits and challenges that closed-ended questions may not capture. By posing these questions, valuable insights can be obtained regarding students' time management methods, helping to identify their preferred strategies and habits. The results from the remaining responses can then be leveraged to assess the effectiveness of these techniques, providing a basis for the development of an application that aligns with, complements, or enhances these existing practices.

Section 3 – Effects of Procrastination and Distraction

Questions 6-10 centred on the impact of procrastination and distractions on respondents. Participants were asked to rate their procrastination and distraction levels on a scale from very regularly to very rarely; the inclusion of these questions aimed to evaluate the effectiveness of the time management strategies identified earlier. Utilizing multiple-choice questions for this purpose facilitated faster completion by users and produced more direct results, which streamlined the analysis process.

Section 4 – Reflection on time management techniques and Applications

Questions 11-14 prompted users to reflect on the time management techniques they had used in the past. By understanding the techniques participants previously employed, it was possible to identify areas where existing solutions may still need to address users' needs fully. This knowledge informed the development of features and functionalities that addressed these gaps and improved existing solutions.

Once the questionnaire design was approved by the project tutor, to make sure the questionnaire adhered to ethical guidelines, ethical approval documents were submitted. After receiving approval from the ethics team, the questionnaire was distributed using Google Forms. This platform was chosen due to its user-friendly distribution method compared to paper-based questionnaires, enabling access to a wider audience through various channels, such as social media and university group chats.

The questionnaire was available for two weeks before the responses were collected. The primary technique to analyze the data was cross-tabulation. The following chapter discusses the findings.

3.2 Findings

After two weeks, the questionnaire garnered 36 responses. The collected data was subsequently cleaned by removing records with inaccurate, inconclusive, or duplicated information. Five responses from individuals who graduated over two years ago and one response with "N/A" as the year of study were excluded, as they contradicted the participant information sheet requiring recent graduates.

The remaining data was structured to facilitate ease of analysis. Following the data cleaning process, a demographic breakdown was conducted. The participant pool exhibited limited age diversity, with 26 respondents aged 18-24, three between 25-34, and one aged 45-54. This lack of age diversity restricts the scope of analysis concerning students' ages. Additionally, there was limited diversity in the participants' year of study: 19 were recent graduates, eight were in their final year, two were in their second year, and one was in their first year. Comparing the experiences of recent graduates to final-year students would provide limited insight, as recent graduates would likely recall their final-year experiences before graduation.

Study subjects were categorized into broader fields: eight participants studied science, ten focused on information technology, eight pursued engineering subjects, and six specialized in business subjects. It is important to note that some participants studied multiple fields, resulting in 32 participants when considering their areas of study.

The results showed that the students used three categories of time management techniques. These included:

3 Task Lists
4 Scheduling
5 Pomodoro technique

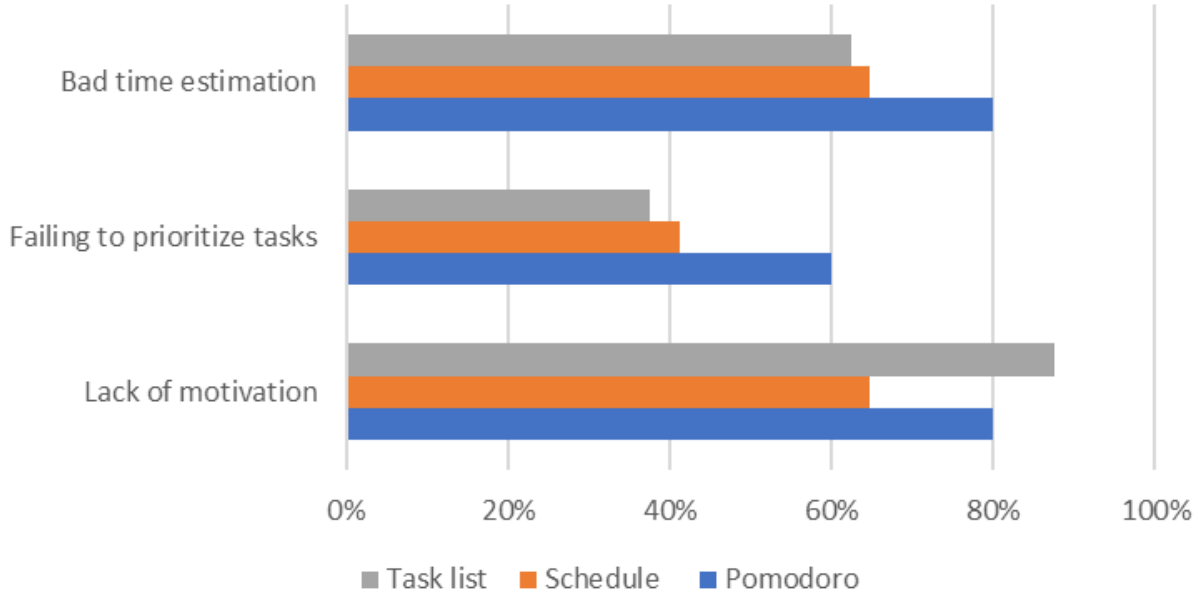


Figure 3.1 – Bar chart displaying what time management issues students had when using certain time management techniques

To identify trends in the data, cross-tabulation was employed in conjunction with responses from other questions. Figure 3.1 illustrates the prevalence of specific time management issues when various time management techniques are utilized. The findings indicate that employing the Pomodoro technique does not significantly impact these factors, as 80% of students reported poor time estimation and lack of motivation. Furthermore, students who used the Pomodoro technique were found to be the least effective in prioritizing tasks. This outcome aligns with the literature review, which suggested that the Pomodoro technique successfully reduces procrastination but may not influence other aspects of time management.

Students who used a task list or a schedule reported similar task prioritization and time estimation results. However, those who relied on a task list experienced a 22.5% higher lack of motivation than those using schedules. 81.8% of all students reported a lack of consistent motivation to study. As highlighted in the literature review, several studies demonstrate that goal-setting can enhance motivation towards goal achievement. Thus, the lack of motivation may stem from an absence of goal-setting within their time management strategies, considering that only 6% of students reported setting goals.

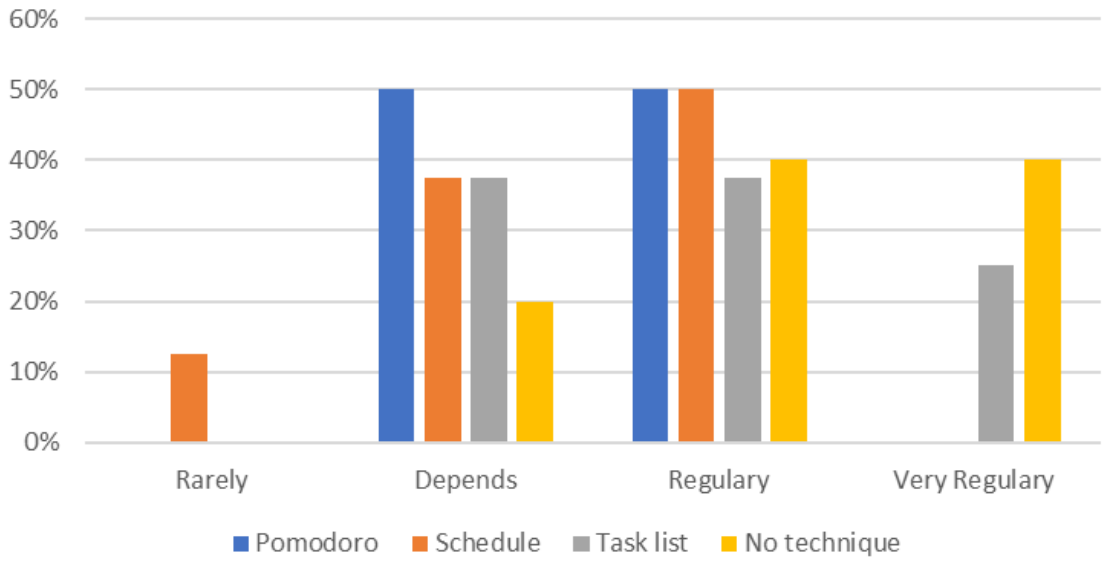


Figure 3.2 – Bar Chart displaying the procrastination levels of different time management techniques

The influence of various time management techniques on procrastination and distraction levels was subsequently examined. Figure 3.2 presents the levels of procrastination associated with different time management techniques. Based on the literature review findings, it was anticipated that students using the Pomodoro technique would exhibit the lowest levels of procrastination. Surprisingly, students utilizing schedules reported marginally lower levels of procrastination.

These results suggest that using task lists has the least impact on reducing procrastination, with 62.5% of students indicating that they procrastinate either 'regularly' or 'very regularly.' Participants were then asked to identify their primary causes of procrastination. This open-ended question yielded responses that mostly fell into the following categories:

- Technology – such as phone notifications and social media
- Boredom
- Other Distractions – such as other people or noisy environments

- Lack of motivation
- Task prioritization

Among students using the Pomodoro technique, 40% cited a lack of motivation as their primary reason for procrastination – the highest among all techniques. However, none of the students utilizing the Pomodoro technique identified distraction as their leading cause of procrastination. Students who employed task lists found technology and other distractions to be their main causes of procrastination. Among schedule users, 38% reported other distractions as their primary reason for procrastination.

The same questions, focusing on distraction, were posed to the participants, with the results illustrated in Figure 3.3. Students using the Pomodoro method reported the lowest distraction level, with 60% responding 'Rarely' or 'Depends'. The responses for schedule users were more varied; 29% indicated they were rarely distracted, while 65% reported being distracted 'regularly' or 'very regularly. Responses from students who used task lists exhibited a similar pattern. Technology and 'other people' were the primary response for all methods on the cause of distraction.

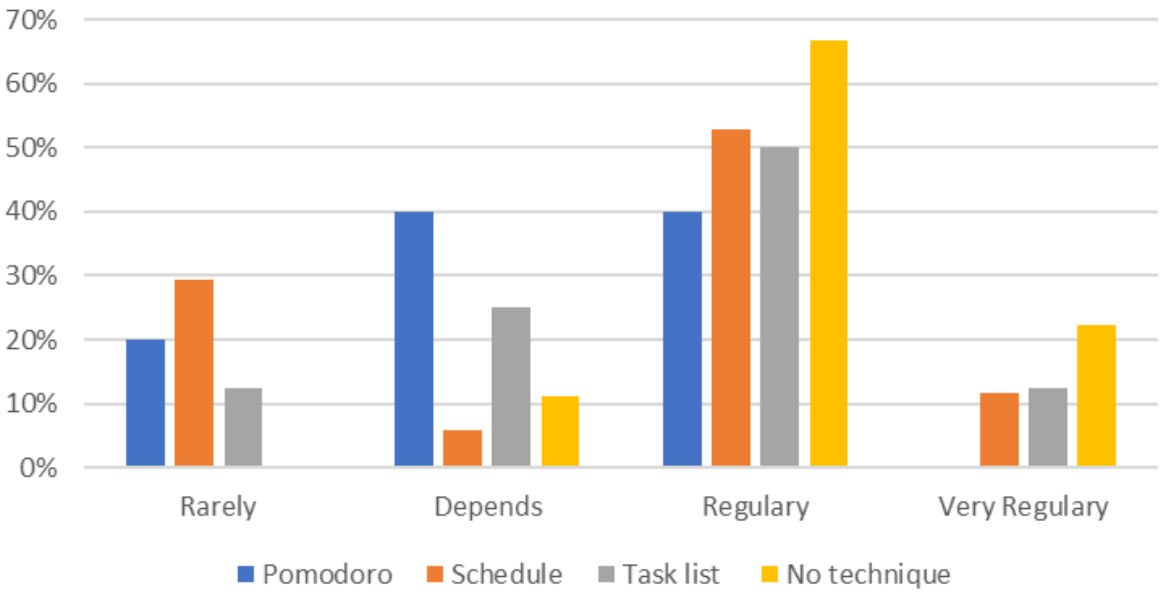


Figure 3.3 – Bar Chart displaying the distraction levels of different time management techniques

As stated in the introduction of this chapter, this knowledge elicitation process aimed to address gaps in the current literature concerning demographic differences in time management. Data on study subjects were therefore analyzed. The initial goal was to consider participants' ages and educational experiences; however, the responses needed to exhibit more variation to identify reliable patterns.

Upon analyzing the data by study subject, it was found that science students reported the highest level of procrastination, with 100% indicating that they procrastinate either 'regularly' or 'very regularly'. In contrast, engineering students' responses were more varied, with 50% responding 'rarely' or 'depends'. Figure 3.4 and Figure 3.5 show each subject's leading causes of procrastination and distraction.

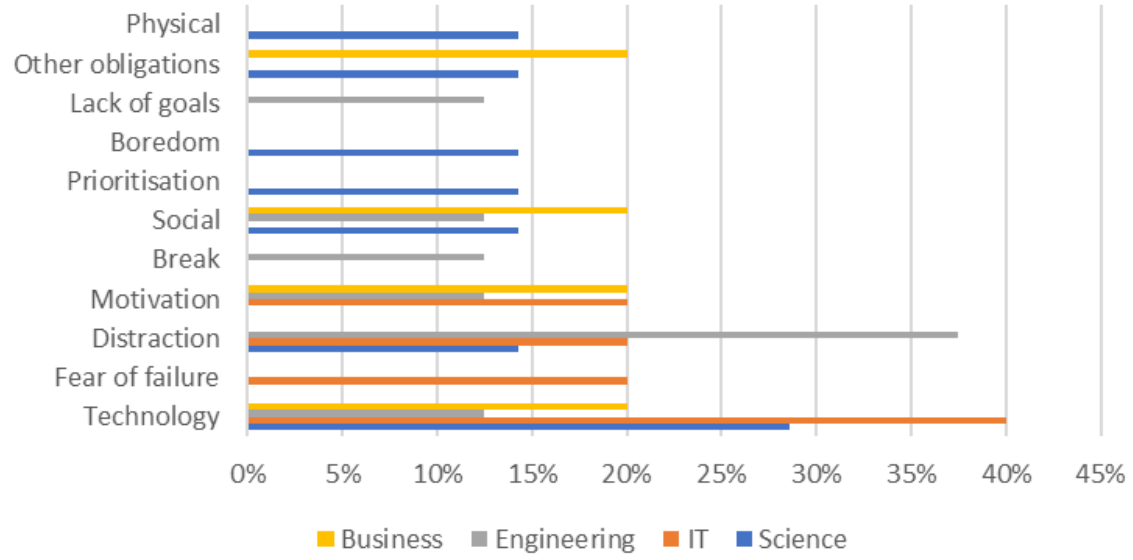


Figure 3.4 – A bar chart showing the causes of procrastination for different subjects

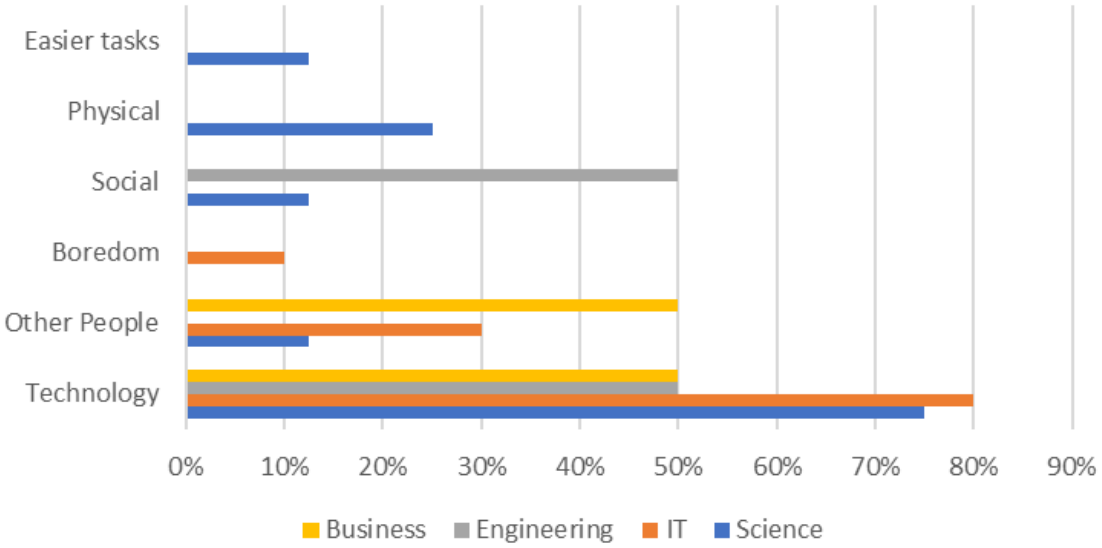


Figure 3.5 – A bar chart showing the causes of distraction for different subjects

The results show that technology emerges as the most significant cause of procrastination and distraction, particularly for IT and science students. One possible explanation could be the inherent technology usage within their studies; for instance, computer science students constantly rely on computers, providing instant access to distracting websites and applications that can sidetrack them from their work. In comparison, business students may be required to engage in more physical tasks. Another noteworthy observation is that the causes of procrastination were more varied than those of distraction.

Business and engineering students identified poor time estimation as the primary impact on their studies, with 83% and 75%, respectively. Conversely, science and IT students reported a lack of motivation as the primary impact, with 86% and 80%, respectively.

The objective for the last two sections was to uncover any patterns related to the success or challenges of a time management technique or software/application. Since most participants have never used or currently do not use a time management software/application, the responses did not provide a clear conclusion, as the answers were unique. However, similar reasons emerged for why students stopped using time management techniques. Students who employed schedules found them either too rigid or ineffective. The recurring issue of students' inability to implement schedules effectively is highlighted in a study by Wajcman (2018), which suggests that a schedule must be well-suited to the user, as one that is too easy is inefficient, while one that is too difficult to follow may increase anxiety. Students who used the Pomodoro technique reported that they were not accomplishing enough work before each break, and using Pomodoro actually led to increased distractions from technology.

3.3 Discussion

The analysis of the questionnaire results aimed to identify the optimal time management technique for a new application. While the Pomodoro Technique effectively reduced procrastination and limited distractions, as supported by Dizon et al. (2021), it fell short in areas such as time estimation, motivation and task prioritization. Consequently, a time management application could benefit from incorporating a Pomodoro timer, but it would also require additional features to address its limitations.

The results revealed that schedules had a comparable impact on reducing procrastination as the Pomodoro timer, and students who used schedules experienced fewer motivation issues. This implies that combining a schedule and a Pomodoro timer might hinder the application's effectiveness, as they share similar effects. It could be argued that a schedule might be more effective since it affects other aspects of time management compared to the Pomodoro timer.

Both task lists and schedules were more effective at prioritizing tasks than the Pomodoro timer. However, nearly 40% of students who used these techniques reported issues with prioritization, suggesting that a time management application should include additional measures to address task prioritization successfully. Task lists and schedules yielded varied results across multiple questions, making it challenging to pinpoint their precise effectiveness. This could indicate that participants did not optimize the use of both methods. Wajcman (2018) emphasizes the importance of utilizing a schedule effectively for success. Therefore, a time management application should communicate this to users and enable them to evaluate the success of their schedules. As students who used task lists faced procrastination issues, a time management application incorporating a task list should also integrate a schedule or Pomodoro timer to mitigate these problems.

The questionnaire revealed that none of the techniques significantly impacted motivation levels, an area in which over 80% of students experienced difficulties. Goal setting could enhance motivation by improving goal attainment (Bahrami Z et al., 2021). However, the questionnaire could not determine the effects of goal setting due to the limited number of respondents using it. As technology was the primary cause of time management issues, incorporating methods to reduce distractions, such as app and website blockers, could significantly improve the application's effectiveness for students.

Although comparing students' subject areas revealed intriguing trends, a more diverse subject group is necessary for a comprehensive conclusion. The results did not include responses from subject areas such as English languages, History, and other languages. It is crucial to consider these disciplines, as their distinct learning and assessment methods could influence the effectiveness of time management techniques for students.

4 Requirements

Trends, common issues and best practices from the previous two chapters are extracted and evaluated to determine the functionalities that users most desire which would be included within this application.

Scheduling and goal setting has been selected as this application's primary time management techniques. The app review indicated that this combination is not commonly used, which sets this application apart. Both knowledge elicitation results and literature suggest that users may lose motivation to adhere to a schedule over the long term, so incorporating goal setting should enhance motivation.

Wajcman (2018) and Tang (2020) emphasize the significance of commitment and review in ensuring the effectiveness of scheduling. Therefore, our application will incorporate features that enable users to assess the effectiveness of their schedules. While some apps have attempted this by providing user analytics, these approaches often fail to encourage users to review and evaluate their progress thoroughly. This application will offer more detailed insights and actionable feedback to increase the uniqueness to help users improve their time management. This approach also addresses the issue identified by Blake et al. (2018) of raising awareness to help users identify and address their time management challenges.

Our literature review underscored the importance of well-defined goals (Covington, 2000). The SMART goal framework will be utilized when guiding users in defining their objectives. Each goal will be assigned a priority level to assist students with poor task prioritization, as the questionnaire results indicated this is a common issue.

Sections 4.1 and 4.2 state the functional and non-functional requirements of the system. Functional requirements define a system or product's essential features and capabilities, specifying its intended behaviour. In contrast, non-functional requirements characterize the overall attributes or qualities of the system or product.

4.1 Functional Requirements

Below are the functional requirements of the system. This defines the needs and goals of the system to guide the design and development process. MOSCOW analysis (Clegg, 2013) was used to prioritize the requirements and objectives of a software system and ensure that the most critical tasks are completed first. The acronym MOSCOW stands for:

- **Must have** - These are critical requirements or features essential for the project to succeed. They are non-negotiable and must be implemented.
- **Should have** - These requirements are important but not critical. Although they add significant value to the project or product, their absence would not render it a failure.
- **Could have** - These requirements or features are desirable but not necessary. If implemented, they are considered "nice to have" and can enhance the project or product. However, their absence would not significantly impact the overall success.
- **Would have** - These requirements or features are the lowest priority and are not planned for the current project or product version.

Table 4.1 displays the functional requirements of this system. Each row defines the requirement number, requirement description, the source of the requirement and the priority of the requirement.

Table 4.1: Table of functional requirements

Number	Functional Requirement	Source	Prioritization
1	System will allow the user to login	App Review + Developer	Must Have
1.1	Users will be able to enter there username and password	App Review + Developer	Must Have
1.1.1	System will allow user to submit their credentials	App Review + Developer	Must Have
1.2	System will alert the user if either the username or password is missing	App Review + Developer	Must Have
1.3	The system will compare the users credentials with every existing record within the database.	Developer	Must Have
1.4	If the users username and password match an existing record within the database, the system should retrieve the users account information.	App Review + Developer	Must Have
1.5	If the users username and password do not match any existing record within the database, the system should alert the user	App Review + Developer	Must Have
1.6	The system should load user's dashboard after successful login	Developer	Must Have
2	System will allow the user to create-an-account	App Review + Developer	Must Have
2.1	User will be able to enter user credentials which include: • Username • Email • Password • Re-enter password	App Review + Developer	Must Have
2.1.1	System will allow user to submit their credentials	App Review + Developer	Must Have
2.2	System will check password and re-entered passwords match	App Review + Developer	Must Have
2.3	If passwords don't match, alert the user	App Review + Developer	Must Have
2.4	System will validate the username, email and password	Developer	Must Have
2.5	If validation is unsuccessful, alert the user	Developer	Must Have
2.6	If the validation is successful and passwords match, the system will enter this information into the UserCredential table	App Review + Developer	Must Have
2.6.1	The system will encrypt the users password	Developer	Should Have
2.7	The system should load user's dashboard after successful account creation	Developer	Must Have

3	The system will have a schedule for each user	Literature review + app review	Must Have
3.1	Users will be able to add events to their schedule	Literature review + app review	Must Have
3.1.1	Users will be able to specify: • Subject • Date • Time • Length • Repetition of each event	App review	Must Have
3.2	The system will repeat these events based on the repetition specified	Developer	Must Have
3.3	The system will allow the user to edit any of the characteristics of the event	Developer	Must Have
3.4	The system will allow the user to delete any event from the system	Literature review + app review	Must Have
3.5	The system will allow the user to add 'Study session onto their schedule'.	Developer	Must Have
3.5.1	The system should allow the user to link every study session with a task	Developer	Must Have
3.5.2	The system should allow the user to edit and delete each of these study session	Developer	Must Have
3.5.3	The system will alert the user if the time block is after the deadline	Developer	Should Have
3.6	The system will store all information on the event and event instances in the database	Developer	Must Have
3.7	The system will allow the user to connect with other calendars	App review	Could Have
3.7.1	The system will incorporate events from other calendars eg 'google calendars' into the users schedule	App review + developer	Could Have
4	The system will allow the users to set goals	Literature review + app review	Must Have
4.1	The system will allow the user to set characteristics of these goals: • Name • Subject • Due date • Priority – out of 10 • Tasks	Literature review + app review	Must Have
4.1.1	The system will allow the user to assign smaller tasks to each goal	App review	Must Have
4.2	The system will allow the user to tell the system when tasks have been completed.	App review	Must Have
4.3	Each goal will be inserted into the goal table within the database	Developer	Must Have

5	The system should all users to enter their subjects	Developer	Must Have
5.1	For each subject the system should allow the user to specify the: • Name • Required weekly hours • Colour	Developer	Must Have
5.2	The system should allow the user to edit and delete subjects	Developer	Must Have
5.3	The system should calculate how much time is spent on each subject within the current week	Developer	Could Have
6	The system will give suggestions on how to improve their time management.	Literature review	Should Have
6.1	The system will allow the user to select time management problems they have been experiencing	Literature review	Should Have
6.1.1	The system will recommend solutions to these problems to the user	Literature review	Should Have
6.2	The system will analyse the users schedule and suggest time management issue	Literature review	Should Have
6.2.1	The system should notify the user if their time is unbalanced	Literature review	Should Have
6.2.1	The system will alert the user if there are any time management issues	Literature review	Should Have

4.2 Non-functional Requirements

Non-functional requirements from this analysis are stated below:

- **Usability** - The app should be user-friendly, with an intuitive interface that is easy to navigate.
- **Security** - User data should be stored securely and protected from unauthorized access, manipulation or disclosure.
- **Compatibility** - The app should be compatible with various devices and platforms, such as iOS, Android, and web browsers.

A comprehensive overview of the front-end and back-end design will be presented in this design chapter. The rationale behind each design choice will be discussed, considering the objectives and requirements of this project.

5.1 Back-End design

5.1.1 Application Type

A web-based platform and an Android application were the considered potential solutions for this time management system. The advantages and disadvantages of each option were thoroughly discussed and analysed, keeping in mind the needs of the students and the project requirements. This comprehensive evaluation aimed to identify the most optimal solution that best caters to the intended audience and aligns with the project's objectives.

Mobile phone usage restrictions in educational institutions can challenge the accessibility of a time management app for students. Due to these constraints, a mobile solution may not be ideal, as students may not be able to use the app as frequently during school hours.

Furthermore, schools and universities often provide desktop equipment, such as computers in libraries and labs, for temporary use by students. It is essential for students to have access to their schedules, goals, and to-do lists wherever they work. In this context, an Android application would be less accessible, as students would not have permission to install it on institutional devices. On the other hand, web-based applications can be accessed from any device with an internet connection and a web browser, regardless of the operating system (Windows, macOS, Linux, Android, iOS). This ensures students can access their schedules, goals, and tasks on any device, offering greater flexibility and convenience.

Moreover, while 70% of mobile applications utilise the Android operating system (Statistica, 2023), an Android application would be inaccessible to 30% of students using other operating systems. By opting for a web-based solution, the deliverable can be a more inclusive and universally accessible time management app for a diverse student population.

However, android applications have greater access to device-specific features such as push notifications and integration with other platforms. This can enable richer and more interactive experiences within a time management app, such as reminders and the device's calendar integration. However, the requirements for this system do not require any device-specific features.

Additionally, mobile applications can function offline, allowing students to access their time management tools and data without an internet connection. This can be particularly useful in areas with limited or unreliable connectivity, ensuring that students can manage their time effectively at all times.

Finally, mobile applications are optimised for specific platforms and can leverage device hardware directly, resulting in faster performance and smoother user experiences. Despite this advantage, the performance difference is negligible as the requirements of this system do not require a large amount of back-end complexity.

Considering the pros and cons of both application types, it was concluded that a web-based application would develop a more accessible application and streamline the development process,

leveraging prior experience. This consideration was crucial due to the constrained timeline of this project.

5.1.2 Development framework

Due diligence on development frameworks was conducted to select the best solution. Both Node.js and Django were evaluated as potential options. Node.js is an open-source, cross-platform runtime environment that allows developers to execute JavaScript code on the server side. It utilises Google's V8 JavaScript engine and features an event-driven, non-blocking I/O model (Node.Js Foundation, 2023). Django is an open-source Python web framework. It follows the Model-View-Controller (MVC) architectural pattern, providing built-in tools for easy database management, user authentication and template rendering. (Django Software Foundation, 2023)

Node js's event-driven architecture can handle real-time updates efficiently. This is essential for many of the requirements of this application. For example, for requirement 3.1, once users create an event, it should update the calendar, which is only possible via real-time updates.

Django provides a built-in authentication system (Django Software Foundation, 2023), which can simplify requirements one and two. Additionally, Django has a powerful templating engine that makes it easy to create dynamic HTML pages (Django Software Foundation, 2023), which can be useful for rendering pages, however, due to Node. Js's extensive range of packages, an application using Node Js could still offer these features, for example, the package EJS allows dynamic front end rendering (EJS, no date).

These frameworks utilise JavaScript and Python, respectively. While Node.js offers superior performance due to its non-blocking, event-driven architecture, the relatively low back-end complexity of the time management application suggests that Django's performance would be sufficient for the project's needs.

Given the limited timeframe available for application development, personal familiarity with each technology significantly influences the decision, as it is impractical to allocate time for learning new technologies. With eight months of professional experience using Node.js and completing the 'Internet Applications and Techniques' and 'Internet Computing' modules, it was determined that Node.js would be the most suitable back-end framework for this project. Becoming proficient with Django's features carries the risk of consuming excessive time, potentially jeopardising the project deadline.

5.1.3 Data Storage

5.1.3.1 Storage Technology

In this time management application, handling structured data is a primary concern. Therefore, a relational database is the ideal choice, as it is designed to store structured data efficiently and effectively manage relationships between entities. Both MySQL and PostgreSQL were considered potential database management systems for this project.

PostgreSQL is renowned for its robust performance when dealing with complex queries (IBM, 2021). However, MySQL was ultimately chosen for its superior performance in read-heavy workloads and straightforward write operations, which aligns more closely with the requirements of this application. Furthermore, the advanced querying capabilities offered by PostgreSQL are not essential for the relatively simple queries needed in this time management application.

5.1.3.2 Required fields

Following the results from the requirements analysis, the data storage requires the capacity to store information in the following categories:

User Information

- Id - A unique identifier for each user to establish relationships between users and their events, goals, subjects, and tasks.
- Name - The full name of the user, used for personalisation and display purposes.
- Email - The user's email address, used for authentication purposes
- Password - The user's encrypted password is used for authentication and maintaining account security.

Event Data

- Id - A unique identifier for each event.
- Date - The date of the event.
- Subject Id - A reference to the associated subject, if applicable.
- Event type - The type of event (e.g., lecture, tutorial).
- Location - The physical or virtual location of the event.
- Start time - The start time of the event.
- End time - The end time of the event.
- Repeat days - The days on which the event repeats, if applicable (e.g., "Monday, Wednesday, Friday").
- Repeat until - The date until which the event repeats, if applicable.
- User id - A reference to the associated user.

Goal Data

- Id- A unique identifier for each goal.
- Name - The name or description of the goal.

- Due date - The date by which the goal should be achieved.
- Priority - The importance or urgency of the goal (e.g., high, medium, low).
- Subject id - A reference to the associated subject, if applicable.
- User id - A reference to the associated user.

Subject Data

- Id - A unique identifier for each subject.
- Name - The name or description of the subject.
- Required Hours - The number of hours allocated to the subject.
- Display Colour - The color used to distinguish the subject in the application visually.
- User id - A reference to the associated user.

Task Data

- Id - A unique identifier for each task.
- Goal id - A reference to the associated goal.
- Name - The name or description of the task.
- Completed - A boolean value indicating whether the task is completed or not.

Task session data

- Id - A unique identifier for each task session.
- Task - A reference to the associated task.
- Start -The start time of the task session.
- End - The end time of the task session.
- Date - The date of the task session.
- User Id - A reference to the associated user.

5.1.3.3 Relationship modelling

An entity relationship diagram was utilised to visually represent the database structure shown in figure 5.1. Each table details each field and its corresponding data type. Additionally, the relationships and multiplicities between each table are demonstrated. Finally, keys are demonstrated with a PK (primary key) or FK (Foreign key) next to the field name.

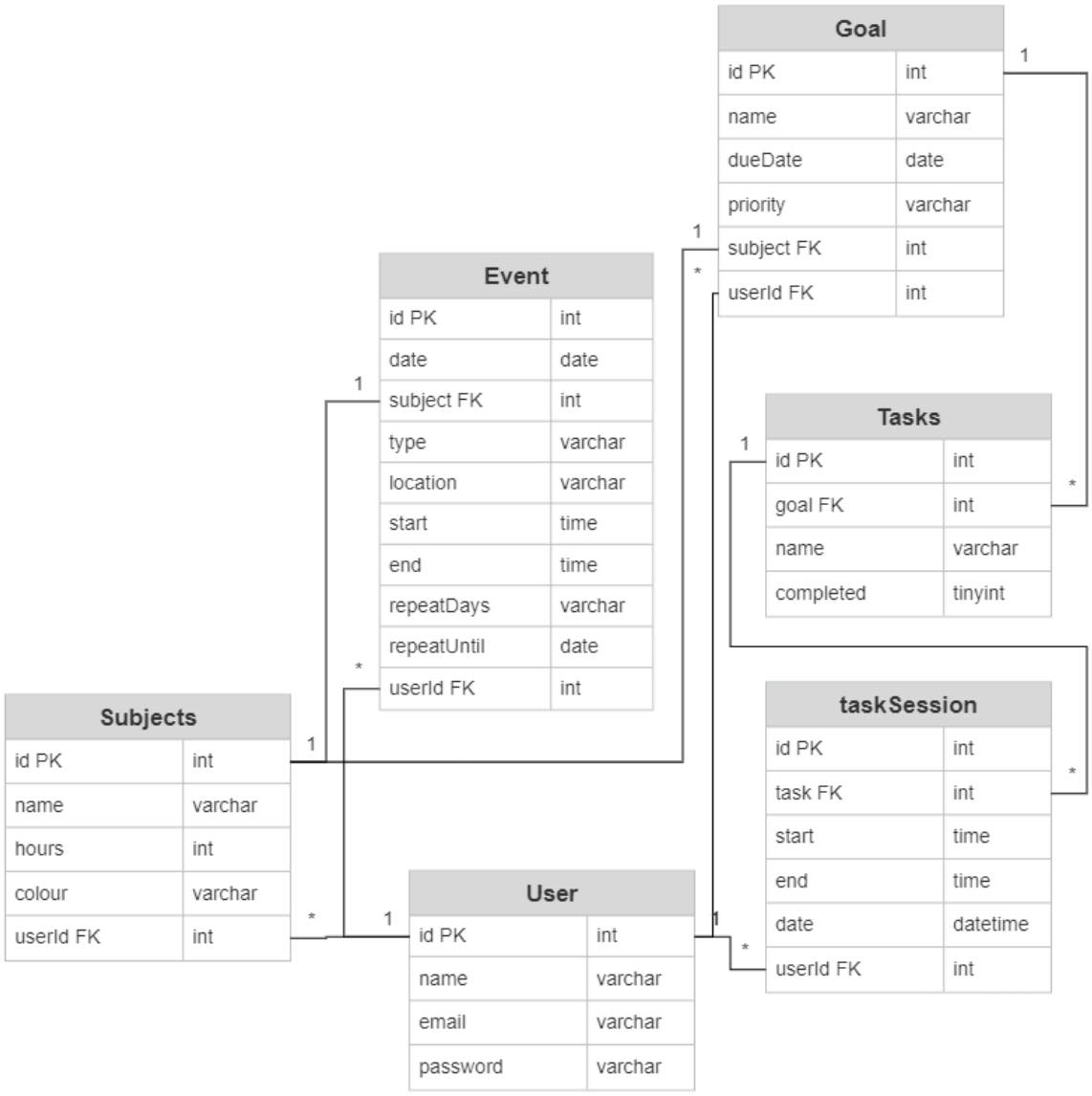


Figure 5.1: Entity relationship diagram for time management application

5.2 Front-End design

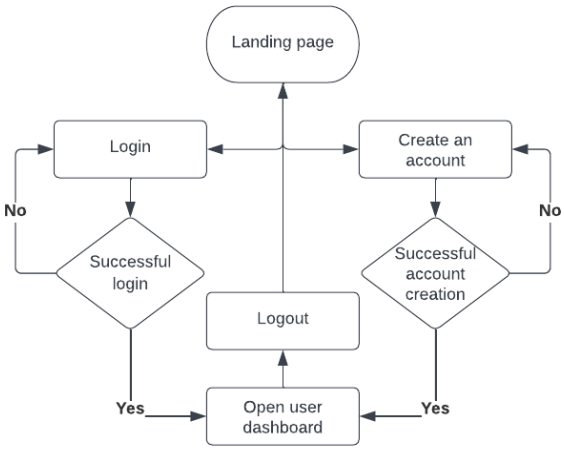
With limited front-end design experience, exploring optimal solutions for the application's user interface was essential. The study materials from the Interaction Design module and Jenifer Tidwell's "Designing

Interfaces" were consulted to achieve this. These resources significantly influenced the design decisions discussed in this chapter.

5.2.1 User interface structure

The initial stage of the front-end design was identifying the structure and organisation of the user interface. The application's overall flow of the application can be identified by utilising a flowchart, as displayed in figure 5.2 and figure 5.3. It was important that the dashboard would contain clear entry points to each dashboard function (Tidwell, 2011). Although figure 5.3 does not demonstrate it successfully, it is also important that each page can go back to the dashboard.

Figure 5.2 – Login System Flowchart diagram



5.2.2 layout

Wireframes were utilised to visually represent the basic structure and layout of the initial design's

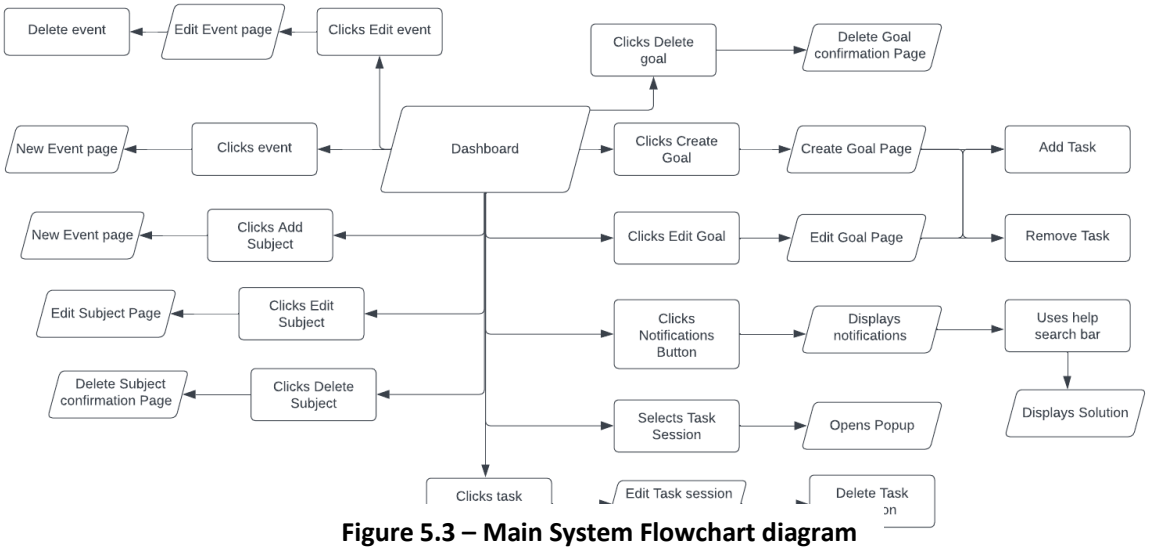


Figure 5.3 – Main System Flowchart diagram

user interface. Wireframes were the tool used to display the layout of each application page. Outside of the dashboard, all other pages had similar layouts which consisted of a form. This chapter

focuses on the design decisions of the dashboard and other forms. The following chapter discusses the decisions made when designing the layout for the application

5.2.1 Forms

The reading pattern for each form is top-to-bottom, as shown in Figures 5.4 and 5.5. Therefore to follow the natural flow, the completion button was placed at the bottom after every input field. Additionally, textual labels were used over icons for these buttons, improving the clarity of the form's function.

Figure 5.4 – Landing Page Wireframe

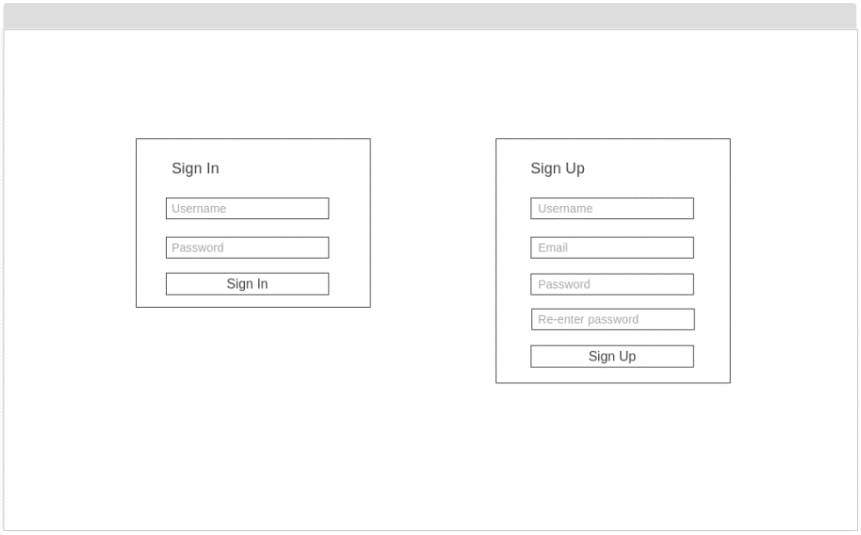


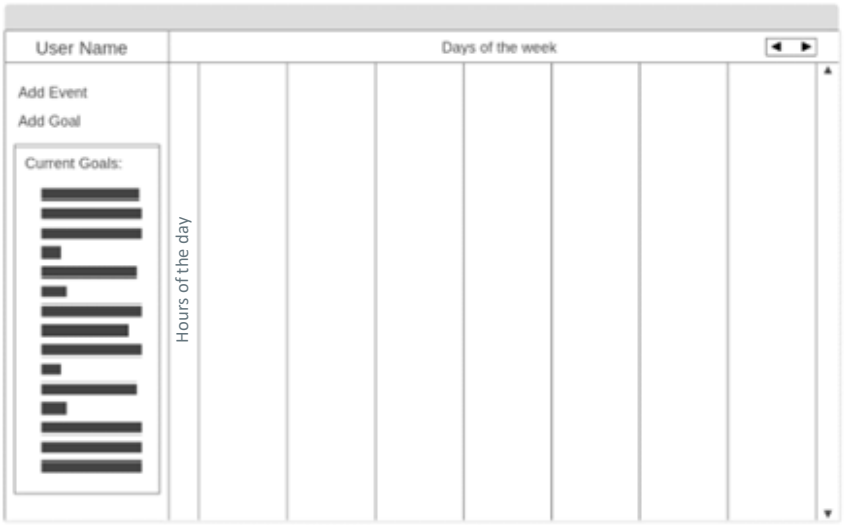
Figure 5.5 – New Goal Wireframe

5.3.2 Dashboard

Upon logging in, the user dashboard is displayed using a tiled panel layout shown in figure 5.6. The tiled panel approach is highly effective for users who want to view multiple elements simultaneously with minimal effort in managing windows. It enables the presentation of numerous features in one central location, which is crucial because it can display the calendar and goals on a single page(Tidwell, 2011). The sidebar contains a list of navigational links, which serve as clear entry points for users to access specific features or functionalities within the application. (Tidwell, 2011)

As discovered in the review of the time management application Sectograph, schedules must be clear to avoid confusion. The schedule, therefore, demands the most space on the page. However, there is a potential drawback to this strategy. Tidwell (2011) indicates that such a layout aligns with the centre stage pattern, establishing a visual hierarchy dominated by the central element. While the schedule is a crucial aspect of the application, the goals are equally important, and this layout might inadvertently shift focus away from the sidebar.

Figure 5.6 – Dashboard Wireframe



5.2.3 Element features

The following chapter highlights the design decisions made outside of the layout.

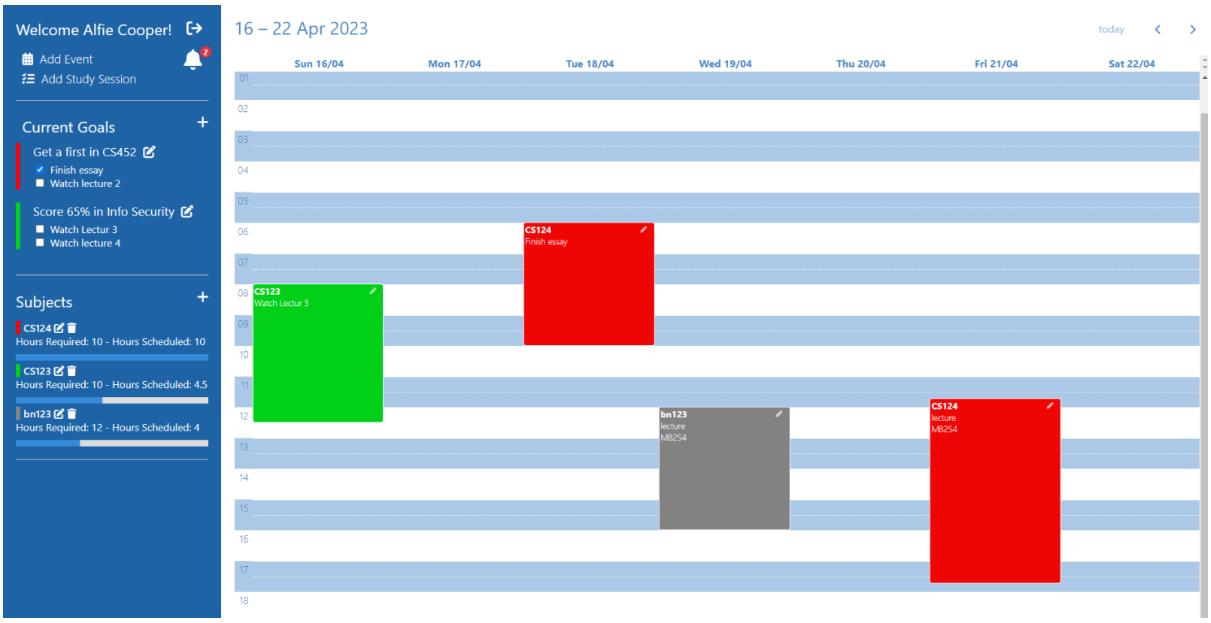
5.2.3.1 Colour

The chosen colour for this project was dark blue (#1d63a5). This was because dark blue is considered a "cool" colour, conveying a sense of respectability and conservatism. White is also a cool colour, and the overall colour scheme would lean towards a cooler and more professional appearance (Tidwell, 2011)

Requirement 5.1 mandates that students can colour-code their subjects, reducing cognitive load by allowing users to rely on visual cues instead of textual descriptions. Figure 5.7 depicts colour coding consistently applied throughout the dashboard to illustrate connections between subject goals and events. One potential issue with this feature is the possibility of overwhelming users with too many colours. To mitigate this, the main body of the application utilises three major hues of the chosen dark blue. This approach aligns with Tidwell (2011), who recommends using only one, two, or at most three major colour hues in the interface, selecting assorted values (levels of brightness) within those hues (Tidwell, 2011). Furthermore, the sidebar and schedule background colours were chosen to create a visual separation between the two components (Tidwell, 2011). Finally, hairlines where utilised to separate different components within the sidebar (Tidwell, 2011).

Red was also used within the main application to alert users. This is shown in Figure 5.7 on the notifications button. Red was used for this as red is a “warm” colour and would, therefore, in contrast with blue, would draw attention and create a strong visual distinction from the rest of the interface (Tidwell, 2011). This can effectively highlight important alerts and make them easily noticeable to users.

Figure 5.7 – Dashboard Display



5.2.3.2 Buttons

All clickable objects on the interface incorporate hover tools, which change the cursor design when hovering over a button. This provides visual feedback to users, indicating that an element is interactive or clickable (Tidwell, 2011).

The Gestalt proximity principle asserts that objects in close proximity will be perceived as belonging (Tidwell, 2011). This concept has been applied throughout the page. For instance, in Figure 5.7, buttons such as the "edit goal" and "edit and delete subject" buttons are placed immediately after the associated component. Additionally, grouped buttons are given the same graphical treatment to indicate connectivity (Tidwell, 2011), as demonstrated by the weekly navigation buttons in Figure 5.7.

5.2.3.3 Text

sans-serif is selected as the main font because it is more readable on computer displays, especially at smaller point sizes (Tidwell, 2011). Pixels may not render tiny serifs well, making sans-serif fonts like these a better choice for body text and label fonts in GUIs (Tidwell, 2011).

Following Tidwell's (2011) assertion on logical hierarchies, various text sizes have been employed to denote the relationship between elements. As illustrated in Figure 5.7, goals are presented with a larger font size than their corresponding tasks. This visual distinction effectively conveys that the tasks are subordinate components of the overarching goal.

5.2.3.4 Icon

Icons were utilised to reduce space consumption. It was made sure that they were all silhouette styles to improve appearance. As shown in Figure 5.7 aggregation was used within the notifications icon to convey a unique message.

5.2.3.5 Input fields

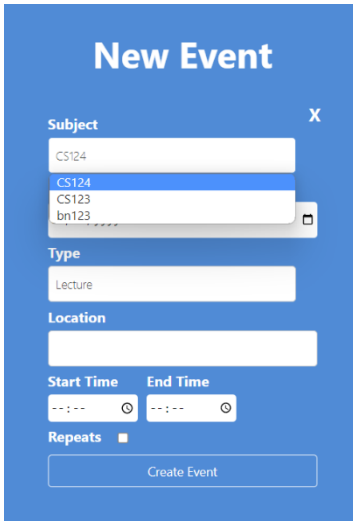


Figure 5.8 – New event form selecting subject

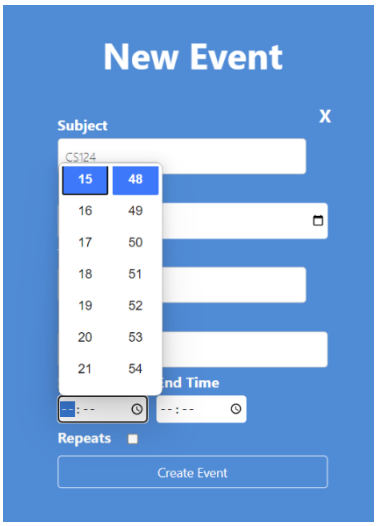


Figure 5.9 – New event form selecting time

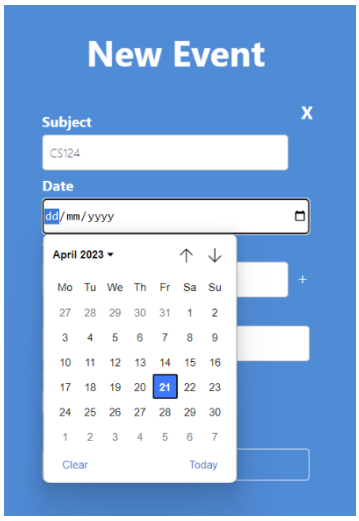


Figure 5.10 – New event form selecting date

Design decisions were made to optimise user input. Combo boxes were used on input with a selected range, such as selecting a subject when creating an event, as shown in Figure 5.8. This is quicker for the user than a separate dialogue box (Tidwell, 2011,p. 351). A structured box was utilised for time inputs,

as shown in Figure 5.9, which ensures the response is in the desired format (Tidwell, 2011,p. 352). The hidden calendar was used for date inputs, which constrained the input to a suitable format, as shown in Figure 5.10. Additionally, having the calendar hidden reduces space consumption (Tidwell, 2011,p. 355).

The repeat days remain concealed to optimise space utilisation until the user activates the "Repeats" checkbox. This design choice ensures efficient use of screen real estate and guides the user to interact with the appropriate input fields. Upon selecting the "Repeats" checkbox, as demonstrated in Figure 5.11, the form expands to reveal the repeat days options.

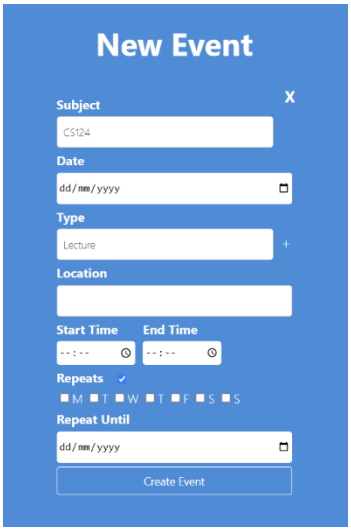


Figure 5.11 New event form selecting repeat days

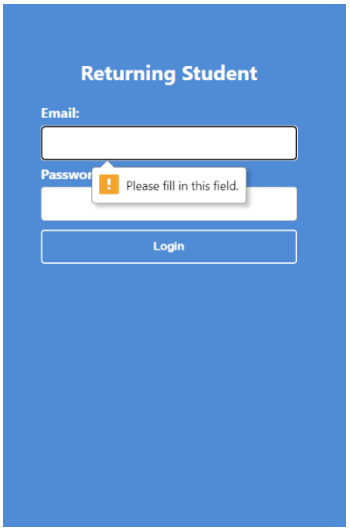


Figure 5.12 Login form submit with empty field

Another input design decision was to keep messages and controls together on the same page so the user could read the message and make the form corrections easily, as shown in Figure 5.12, with no jumping around or error-prone memorisation (Tidwell, 2011).

6 Development

Prior to development, due diligence relating to development options was completed. Development procedures, frameworks and tools were researched, allowing for the most suitable selection. This informed a development plan which ensured that the chosen solutions aligned with the project's goals and requirements, maximising efficiency and productivity

6.1 Software lifecycle

The software development lifecycle impacts many aspects of the development process; therefore, it was essential to select the appropriate lifecycle to maximise the project's success. Both the agile and waterfall approach was considered.

6.1.1 Waterfall Model

The Waterfall Software Development Life Cycle model is a linear approach to software development. It is characterised by a sequential flow of stages, with each stage depending on the successful completion of the previous one (Royce, 1970). The waterfall model typically consists of the following stages:

- **Requirements Gathering and Analysis:** In this phase, all potential requirements for the system under development are collected and documented in a requirements specification document**.**
- **System Design:** The requirements from the first phase are examined during this stage, and a system design is created. This design outlines hardware and system requirements and defines the overall system architecture.
- **Implementation:** Developers write the code for the software system based on the design specifications. This involves programming, unit testing, and integration of different modules.
- **Testing:** Once the implementation is complete, the software system undergoes rigorous testing to ensure that it meets the requirements and is free of defects. Testing typically involves functional tests, performance tests, and user acceptance tests.
- **Deployment:** After successful testing, the software is deployed to the production environment, making it available to end users.
- **Maintenance:** The software system is monitored and maintained to address any issues, bugs, or enhancements needed in the future.

This method presents advantages and disadvantages that can impact the project outcome. Advantages of the Waterfall method include its simplicity, as it follows a linear, sequential process with clearly defined stages that are easy to understand. Each stage has specific deliverables providing structure, ensuring the project stays on track and maintains quality. The linear structure and clear milestones make planning and monitoring progress easier. The Waterfall model is particularly suitable for small projects with clear requirements and minimal changes (Royce, 1970).

On the other hand, the Waterfall method has its share of disadvantages. Its inflexibility means it is not well-suited for projects with changing requirements or evolving needs, as it is difficult to accommodate changes once the development process has started. Furthermore, end users typically provide feedback only after the system is complete, making it time-consuming to address any issues. Testing is performed late in the development process, which can lead to the discovery of defects at a later stage, making them more expensive and difficult to fix. Lastly, the Waterfall model does not promote frequent communication with the end users, which may result in a final product that does not meet their expectations or needs (Royce, 1970).

6.1.2 Agile

Agile development methodology is a modern, iterative approach to software development that emphasises flexibility, collaboration and customer satisfaction. This approach breaks down projects into small increments, enabling continuous improvement and adaptation to changes. Agile promotes close collaboration among cross-functional team members and prioritises regular communication and feedback loops. With adaptive planning processes, Agile allows for frequent revisions based on new information or changes in scope. The methodology is focused on the early and continuous delivery of value, ensuring that the final product meets the needs and expectations of end users (Beck et al., 2001).

Flexibility is a key advantage of Agile, as its iterative approach allows teams to adapt easily to changes in requirements and priorities. This makes Agile suitable for projects with evolving needs or uncertain scopes. Another benefit is the emphasis on collaboration, which encourages close interaction among cross-functional team members and stakeholders. This results in improved communication, a better understanding of requirements, and more efficient problem-solving. Agile also prioritises early and continuous delivery of working software, enabling end users to provide feedback and derive value from the product sooner. The iterative nature of Agile also aids in risk management, as it allows for early identification and mitigation of risks, preventing issues from escalating later in the development process. Lastly, Agile's focus on frequent testing and continuous integration enhances software quality (Beck et al., 2001).

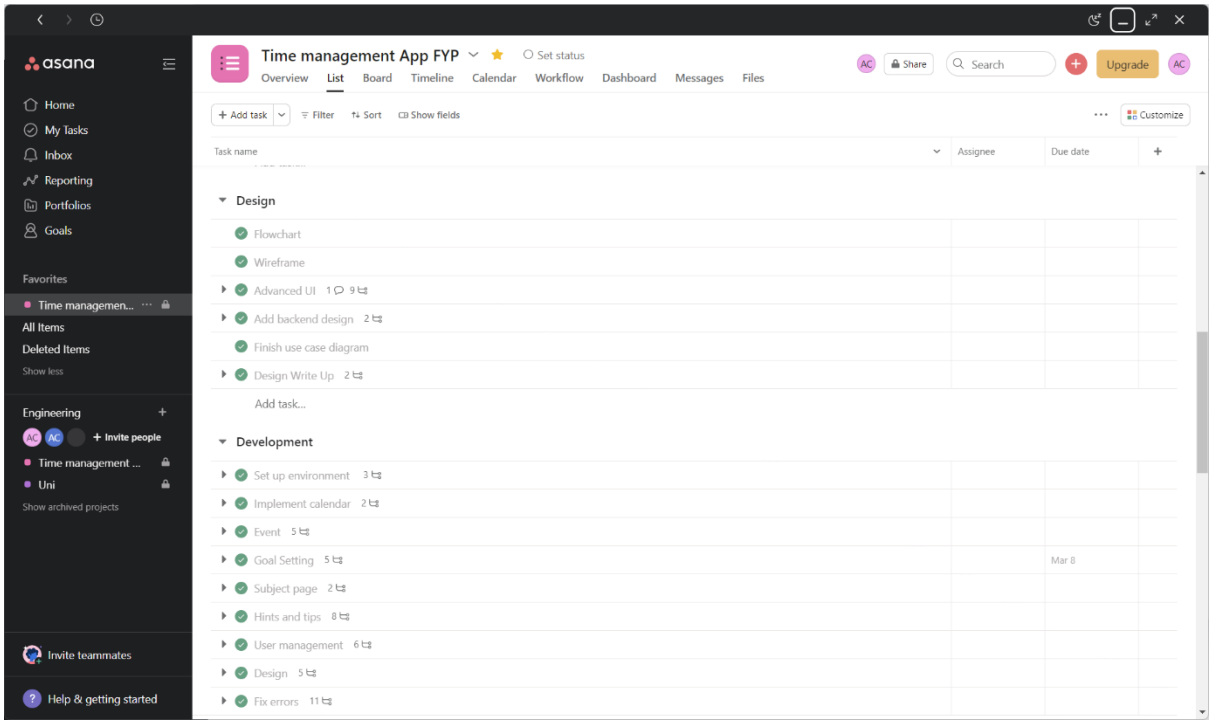
However, Agile does present some challenges. Its adaptive planning process can make it difficult to predict the exact timeline of a project. The flexibility of Agile methodologies can sometimes result in scope creep, where project goals and requirements expand beyond the initial plan, causing delays. (Beck et al., 2001).

6.1.3 Discussion

Before initiating the development process, the Waterfall methodology was chosen as the preferred software development lifecycle due to the stable nature of the requirements and the need for accurate timeline estimation (Royce, 1970). This choice was particularly important for this project, as it had a strict deadline, and the risk of scope creep had to be minimised. To successfully implement the Waterfall method, Asana was employed as the project management tool for documenting and managing each phase/

Figure 6.1 displays the Asana dashboard, showcasing the design and development phases. The corresponding tasks, sub-tasks, and notes were added for every phase to ensure comprehensive documentation and organisation. A thorough review was conducted upon completing all the items in a specific phase. The associated tasks and deadlines were updated if any incomplete tasks were identified. Conversely, if the phase was deemed complete, the project advanced to the next phase, maintaining a structured and sequential approach throughout the development process.

Figure 6.1: Asana Dashboard



However, upon the usability testing phase, changes in design were required, which meant the project tilted towards more of an agile approach. Figure 6.2, displays the adapted waterfall methodology for this project.

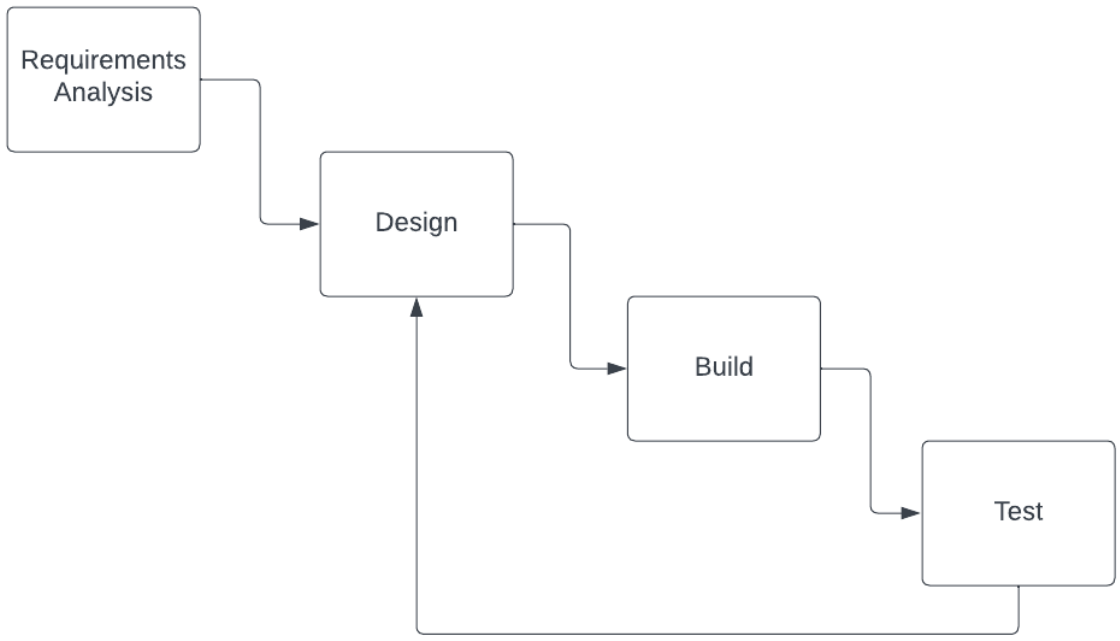


Figure 6.2: Projects software lifecycle methodology (adapted from Beck et al, 2001)

6.2 Development tools

This section will detail the development tools required to complete this project and optimise this application.

6.2.1 Application deployment

Web applications typically run on hosted external servers (Fielding & Taylor, 2002); however, deploying the application on a local machine for this project's development offered several advantages. These advantages included the elimination of dependency on a stable internet connection for development, allowing for a more seamless workflow. Furthermore, using a local machine enabled faster testing and debugging. After each change, developers could directly modify the codebase and immediately observe the effects without deploying the code to a remote server. This approach significantly reduced time spent on iterative development tasks and facilitated quicker manual testing.

While deploying the application on another server might have provided some advantages, such as more closely simulating a production environment, it was ultimately decided that the application would be deployed on a local machine during the development phase. To achieve this, a development environment compatible with Node.js and MySQL was required, allowing the application to be accessed through localhost.

Various solutions, including XAMPP and WAMP, were evaluated, as both offered the benefits of Apache's HTTP Server and MySQL configuration, making them suitable for running the application. Ultimately, Xampp was selected due to prior development experience, which minimised configuration time and leveraged familiarity with its folder structure and features

6.2.2 Plug-ins and Packages

Several plug-ins packages were employed to enhance the capabilities of Node.js and JavaScript for the successful completion of this project. These additional tools augment the core functionality and facilitate the development process while ensuring that the project requirements are met effectively and efficiently.

For requirement 3, using a plug-in was deemed the most effective strategy for implementing the schedule to mitigate the risk of development delays. Coding a complex feature like a schedule from scratch can be time-consuming and affect other components' development. FullCalendar and DayPilot were considered, as they were capable of meeting the system's requirements. FullCalendar offered advantages such as time selection and dynamic data loading (fullCalendar, 2023), while DayPilot provided multiple components, including a calendar, scheduler and Gantt chart (DayPilot, 2023). Ultimately, FullCalendar was chosen because it offered features that DayPilot did not, such as time selection (FullCalendar, 2023), which is crucial for requirement 3.5. Additionally, the lite version of DayPilot has limited features compared to FullCalendar (DayPilot, 2023).

Requirement 2.6.1 specified that passwords must be encrypted. The decision was made to utilise a plug-in, specifically bcrypt (bcrypt, 2021), as this feature must sufficiently work and be tested to maintain security. Bcrypt is a widely-used password hashing library that dynamically adjusts the hashing cost according to hardware capabilities. As computers become more powerful, the cost factor can be increased to maintain a high level of security against brute-force attacks. Furthermore, bcrypt automatically generates a unique salt for each password, protecting against precomputed dictionary attacks, also known as rainbow table attacks (bcrypt, 2021). Given that bcrypt met all requirements for this system, alternative solutions were not explored.

Additional middleware was required, such as Body-parser, a middleware that parses incoming request bodies in a middleware before the handlers, making it easier to handle and process data submitted by users via forms (body-parser, 2020), and cookie-parser, a middleware that parses cookies attached to client requests, facilitating the management of user sessions and authentication (cookie-parser,2022).

As JavaScript or EJS does not include any technique for adding icons, an icon library was included. FontAwesome was chosen as it had all the icons needed within the application's design (FontAwesome, 2023).

The library Moment was used for working with dates and times, simplifying the application's manipulation, formatting, and parsing of date and time data (Moment.js, 2023). This was needed to format the due dates of the goals for requirement 3.5.

For requirement 5.1.1, it was decided that a natural language model would be used to select the right solution to the user input. The plug-in Natural was required to aid in tasks like tokenisation, stemming and classification (natural, 2023).

6.1.5 Integrated development environment

In selecting an appropriate IDE for this project, Visual Studio Code emerged as the optimal choice due to its robust Git integration capabilities and extensive support for a diverse range of languages, frameworks, and libraries, which are crucial for an application that necessitates various resources. Although other IDEs, such as WebStorm and Atom, were evaluated, familiarity played a significant role in the decision-making process. Ultimately, Visual Studio Code was deemed the most suitable platform to facilitate efficient and effective development for this project.

6.2 Code standards

During development, code standards were established, attempting to maintain overall code quality and readability: this section lists and evidences code standards used throughout the project.

6.2.1 Modularity

Complex code was broken down into smaller reusable functions, and different concerns were separated into different files. The code was organised into separate files based on functionality. Figure 6.3 displays part of the function that loads the dashboard. This code demonstrates how database operations and data formatting concerns have been separated by leveraging helper functions from separate module 'formatHelpers'.

Figure 6.3: Load dashboard function code

```
// Route handler to load the dashboard page
exports.loadDashboard = async (req, res) => {
  try {
    // Get user ID from cookies
    const userId = req.cookies.userId;

    // Perform multiple database queries in parallel using Promise.all
    const [
      subjects,
      eventResult,
      taskSessionRes,
      goalsResults,
      tasksResults,
      username
    ] = await Promise.all([
      query('SELECT * FROM subjects WHERE userId=?', [userId]),
      query('SELECT * FROM event WHERE userId = ?', [userId]),
      query('SELECT * FROM tasksession'),
      query('SELECT * FROM goals'),
      query('SELECT * FROM tasks'),
      query('SELECT name FROM user WHERE id=?', userId)
    ]);

    // Merge the event data with the subject data
    const mergedArray = formatHelpers.mergeEventDataWithSubjectData(eventResult, subjects);
    // Format the event data for display in the calendar
    const events = formatHelpers.formatEventDataForCalendar(mergedArray);
    // Format the task session data for display in the calendar
    const taskSessions = formatHelpers.formatTaskSessionsForCalendar(taskSessionRes, tasksResults, goalsResults, subjects);
    // Update subjects with accumulated time
    const updatedSubjects = formatHelpers.updateSubjectsWithAccumulatedTime(subjects, events, taskSessions, tasksResults, goalsResults);
    // Calculate the days remaining until the deadline
    const daysToDeadline = [];
    // Generate messages based on goal results, updated subjects, task sessions, and task results
    const messages = formatHelpers.generateMessages(goalsResults, updatedSubjects, taskSessions, tasksResults);
```

6.2.2 Promises and async/await

Promises and async/wait is utilised to manage functions that do not complete immediately, such as database queries. Figure 6.3 demonstrates how a promise was used to retrieve database information.

6.2.3 Error Handling

Error handling was implemented to identify, isolate, and fix issues during development. By adding this code, the source of the error and how to address the error is more clear. Figure 6.4 demonstrates this by including a try/catch block. This will attempt to run the code within the try block, if unsuccessful, the error will be caught and displayed on the terminal.

Figure 6.4: Insert subject into database function

```
exports.addSubject = async (req, res) => {
    const sql = 'INSERT INTO subjects (name, hours, colour, userId) VALUES (?)';
    const name = req.body.newEventSub;
    const hours = req.body.hours || 0;
    const colour = req.body.color;
    const user = req.cookies.userId;
    const dbInsert = [name, hours, colour, user];

    try {
      await db.query(sql, [dbInsert]);
      res.redirect('/dashboard');
    } catch (err) {
      console.log(err);
    }
  };
```

6.2.4 Environment variables

Environment variables were utilised to store sensitive information, such as database credentials. Figure 6.5 displays how environment variables were used to store the user's database credentials. This means that malicious users cannot obtain access to the user's password with just the source code.

Figure 6.5: Database connection code using environment variables

```
// Create a connection to the database using environment variables
const db = mysql.createConnection({
  host: process.env.DATABASE_HOST,
  user: process.env.DATABASE_USER,
  password: process.env.DATABASE_PASSWORD,
  database: process.env.DATABASE,
});
```

6.2.5 Comments

As displayed in Figure 6.3, comments were used throughout development to explain the purpose and functionality of each part of the code.

6.3 Development Process

The development was conducted through weekly stages. Tutor meetings every Monday allowed subsequent feedback on the feedback of project where any necessary adjustments to the requirements. The following details the significant events and milestones achieved during each week:

6.3.1 Week 1

The environment setup was straightforward; however, it took some time to become acquainted with all aspects and reacquaint certain elements. The calendar implementation progressed smoothly, as most of the research was conducted during due diligence.

6.3.2 Weeks 2-4

Basic functionality for goal setting, events, and subjects was successfully implemented. However, following a review by the module tutor, it was suggested that the application needed more complex features to enhance the project's unique selling proposition (USP). Although this change introduced some delays in development, the incorporation of new requirements ultimately resulted in a significantly improved application.

6.3.3 Weeks 5-6

The primary objective of these weeks was to implement requirement 5, which represented one of the most challenging phases of development. Due to delays in previous weeks, timely completion of this requirement was crucial. The complexity of these requirements, including the data analysis functions, added to the challenge of these weeks. An example of a particularly challenging function was the method that calculated the time spent on each goal in Figure 6.7. This was challenging, considering the data formatting required to perform the calculations to get the correct result.

```
function timeOnEachGoal(goals,tasks,taskSession){
  // First, create an object to store the time spent on each goal
  const timeSpentPerGoal = {};

  // Iterate through the taskSession array to calculate the time spent on each task session
  taskSession.forEach(session => {
      const task = tasks.find(t => t.id == session.taskId);
      console.log(session, tasks)
      const goal = goals.find(g => g.id == task.goal);
      const timeDiffMs = Date.parse(`1970-01-01T${session.end}Z`) - Date.parse(`1970-01-01T${session.start}Z`);
      const timeDiffMin = timeDiffMs / (1000 * 60);

      if (!timeSpentPerGoal[goal.id]) {
          timeSpentPerGoal[goal.id] = 0;
      }

      timeSpentPerGoal[goal.id] += timeDiffMin;
      });

      // Create a new array of goals with the extra timeSpent field
      const goalsWithTimeSpent = goals.map(goal => {
      const goalTime = Object.entries(timeSpentPerGoal)
          .filter(([goalId]) => goalId === goal.id.toString())
          .reduce((total, [, time]) => total + time, 0);

      return {
          ...goal,
          timeSpent: goalTime
      };
      });

      return goalsWithTimeSpent;

}
```

Figure 6.6: Function to calculate the time on each goal

For requirement 6, it was determined that a natural language processor (NLP) would be the optimal solution. This part became particularly complex because there was no prior experience working with language processors. Figure 6.7 displays the function to process the input; it became challenging due to inexperience working with tokenisation and language models. However, the 'Natural' package and its accompanying documentation (Natural, 2023) successfully facilitated the completion of requirements six on schedule. Completing requirement 6 in the time frame was a notable achievement as it was the most complex part of development.

```
// Function to process user input
function processInput(input) {
  const tokens = tokenizer.tokenize(input);
  return tokens.map(token => stemmer.stem(token.toLowerCase()));
}

function findBestMatchingIssues(input) {
  const processedInput = processInput(input);
  const uniqueIssues = new Set(processedInput);
  let bestMatches = [];

  uniqueIssues.forEach((issueWord) => {
    let bestMatch = { index: -1, score: 0 };
    issues.forEach((issue, index) => {
      const processedIssue = processInput(issue.issue);
      const intersection = processedIssue.filter(token => token === issueWord);
      const score = intersection.length / processedIssue.length;
      if (score > bestMatch.score) {
        bestMatch = { index, score };
      }
    });

    if (bestMatch.index !== -1) {
      bestMatches.push(issues[bestMatch.index]);
    }
  });

  return bestMatches;
}
```

Figure 6.7: Function to process user input to find the time management solution

6.3.4 Week 7

During this sprint, the primary objective was integrating user management functionality into the application and incorporating the specified design elements. This implementation proved to be more challenging than anticipated. This oversight resulted from its late addition, necessitating the modification of existing functions.

6.3.5 Week 8

Due to impending deadlines from other modules, the objectives for this sprint were scaled back to accommodate the limited time available. Nonetheless, progress in implementing the CSS was successful.

6.3.6 Week 9

During this sprint, all design and functional requirements were successfully fulfilled, culminating in creating of a minimum viable product for the project. In addition, a usability testing plan was developed, accompanied by all necessary documentation to secure ethical approval.

6.3.7 Week 10

Upon obtaining ethical approval, the usability testing process was initiated. The usability tests were efficient and successful because the testing plan was already in place, and participants had been pre-

selected. As this marked the final scheduled development week, ensuring the completion of both the redesign and development within the stipulated time frame were imperative.

6.4 Conclusion

Throughout the development process, careful planning, research, and decision-making were conducted to ensure the success of the project. Various development tools, plugins, and packages were utilized to enhance the application's capabilities and streamline the development process. Code standards were established and maintained to ensure code quality and readability.

The development process of the project unfolded through a series of weekly stages, with each week focusing on specific objectives and milestones. Regular meetings with the tutor allowed for constant feedback and any necessary adjustments to the requirements. The environment setup, implementation of basic functionality, and integration of more complex features such as data analysis functions and natural language processing contributed to the project's unique selling proposition (USP).

Challenges and delays were encountered throughout the development process. The integration of user management and design elements proved to be more complex than anticipated, but progress was made despite limited time availability.

7 Testing

This chapter examines the various testing methodologies employed throughout the project, including automated, manual, and usability testing. The primary objective of this testing phase is to assess the application's functionality, identify potential issues, and make the necessary improvements to optimise the user experience.

7.1 Automated Testing

Unit testing was used for the method of automated testing. Unit testing is a software testing methodology in which individual components or units of a software application are tested in isolation to ensure they function correctly. A unit is the smallest testable part of a software system and typically comprises a single function, method, or class (Osherove, 2009).

This application primarily consists of interconnected components, with the main components being requirements 3 and 4. Moreover, the project includes numerous individual helper functions due to the emphasis on modularity, as specified in section 6.2.1. Unit testing was selected because it helps ensure that each component and function operates correctly in isolation before being integrated with other components.

To implement these unit tests, the Mocha testing framework was utilised. Mocha is designed for testing asynchronous and synchronous code in Node.js and browser-based applications (MochaJS Contributors, 2022). Mocha was chosen due to the team's prior experience with this technology.

All unit tests were carried out in a separate file named test.js. The initial plan was to incorporate unit tests throughout the entire project. However, creating unit tests for larger components, such as the schedule, proved more challenging than anticipated. Consequently, manual testing was deemed more feasible for requirements 3 and 4. Nevertheless, unit testing was still implemented for the helper functions.

Figure 7.2 illustrates a unit test for the helper function checkPriorityTimeSpent, which is shown in Figure 7.1. This function analyses the time spent on each goal and returns an error message if a goal with a lower priority has more time allocated than one with a higher priority. The code in Figure 7.2 conducts two tests: the first test examines goals with an error, while the second investigates goals without errors. Figure 7.3 presents the results of these tests, demonstrating that the output matched the expected outcomes.

Figure 7.1: Test function

```
function checkPriorityTimeSpent(goals) {
  messages=[]
  let error = null;
  goals.forEach((goal1) => {
     goals.forEach((goal2) => {
       if (parseInt(goal1.priority) < parseInt(goal2.priority) && goal1.timeSpent < goal2.timeSpent) {
         error = `More time spent on lower priority goal '${goal2.name}' than on higher priority goal '${goal1.name}'`;
         messages.push(error);
       }
     });
   });

  return messages;
}
```

Figure 7.2: Unit tests

```
//----------Unit test 2 - test
describe('checkPriorityTimeSpent', () => {
    it('should return an error message if more time is spent on a lower priority goal than on a higher priority goal', () => {
      const goals = [      { name: 'Goal A', priority: '4', timeSpent: 10 },      { name: 'Goal B', priority: '3', timeSpent: 5 },
      { name: 'Goal C', priority: '2', timeSpent: 12 },      { name: 'Goal D', priority: '1', timeSpent: 13 }    ];
      const expectedOutput = ["More time spent on lower priority goal 'Goal A' than on higher priority goal 'Goal B'"];

      const result = checkPriorityTimeSpent(goals);

      assert.deepStrictEqual(result, expectedOutput);
    });

    it('should return an empty array if no errors are found', () => {
      const goals = [      { name: 'Goal A', priority: '4', timeSpent: 10 },      { name: 'Goal B', priority: '3', timeSpent: 15 },
      { name: 'Goal C', priority: '2', timeSpent: 20 },      { name: 'Goal D', priority: '1', timeSpent: 25 }    ];
      const expectedOutput = [];

      const result = checkPriorityTimeSpent(goals);

      assert.deepStrictEqual(result, expectedOutput);
    });
  });
```

Figure 7.3: Unit test output

```
  checkPriorityTimeSpent
    ✓ should return an error message if more time is spent on a lower priority goal than on a higher priority goal
    ✓ should return an empty array if no errors are found
```

7.2 Manual Testing

Manual software testing involves human testers evaluating the quality of a newly developed application by executing test cases without the use of automated tools or scripts (Kaner, Falk & Nguyen, 1999). Black box testing was used for manual testing. Black box testing is a method that validates the software's functionality without delving into its internal structure or implementation. This approach emulates the end user's perspective, ensuring that the application operates as intended and offers a user-friendly experience (Pressman & Maxim, 2014).

Use cases in black-box testing play a vital role in ensuring that software meets its functional requirements by providing a user-centric perspective and simulating real-world scenarios that end users are likely to experience (Pressman & Maxim, 2014). In this project, black-box testing was applied in functionality and integration testing to ensure all components tested by the unit tests integrate successfully, focusing on tasks such as creating, editing, and deleting events, goals, and subjects, as well as user account management. Test cases were defined using behaviour-driven development user stories, describing a scenario along with the expected outcome. The actual outcome was then compared with the expected outcome to determine if the test passed or failed, as illustrated in Table 6.1, which presents use cases for requirement 1.

Scenario	Expected outcome	Result	Pass/fail
Given that I am a returning student and I want to login when I enter my credentials on the landing page and press sign in. Then my dashboard should be displayed with my goals, subject and schedule.	The dashboard should be loaded displaying the user's goals, subjects and schedule	The dashboard was loaded, displaying the user's goals, subjects and schedule	Pass
Given that I am not returning student and I want to login when I enter my credentials on the landing page and press sign in. Then an error should be displayed	An error message should be displayed and the system should remain on the landing page	An error message was displayed and the system remained on the landing page	Pass

Table 6.1: Login Use Case

7.3 Usability Testing

Usability testing is a user-centred evaluation method involving real users interacting with a product or system to identify issues and enhance the overall user experience (Pressman & Maxim, 2014). This chapter presents the objectives, target audience, and methodology of the usability testing conducted on the application.

7.3.1 Objectives

1. The primary objectives of this usability testing were to:
2. Assess the ease of use and visual clarity of the calendar.
3. Evaluate the user experience when creating, editing, and deleting events, goals, and subjects.
4. Verify the correct display and integration of events with the calendar and related subjects.
5. Assess the integration of goals with tasks and other features, such as their display on the sidebar.
6. Test the user interaction by scheduling study sessions using the calendar.
7. Evaluate the integration of study sessions with corresponding goals.
8. Assess the effectiveness of notifications in highlighting schedule optimisation opportunities.

The application is designed for a diverse range of users who need time management and scheduling assistance. Five willing participants were selected for the usability testing, and their identities were kept anonymous for ethical reasons

7.3.2 Methodology

The following methodology was employed to conduct the usability testing, ensuring ethical considerations were met:

1. Participants received a Participant Information Sheet and Consent form before the usability test meeting, which they must sign and return.
2. The usability test was conducted via Microsoft Teams, which allowed screen recording and remote control of the researcher's screen by the participant.
3. The screen was shared, and participants were asked to request control to interact with the app.
4. Participants were provided with dummy data to work with, ensuring the protection of personal information.
5. Participants were asked to complete prescribed tasks using the app while 'thinking aloud.' The researcher made observation notes in addition to the participants' utterances.
6. After completing the tasks, participants were asked a series of questions to further probe their opinions on the app's strengths and weaknesses.
7. Audio/video recordings of each session were created and stored securely on an encrypted cloud server. These recordings were used to create anonymised content logs and transcripts. Once completed, the recordings were securely destroyed to protect participants' confidentiality and anonymity.
8. Quantitative measures, such as task completion time and number of taps, as well as qualitative measures based on participants' comments and answers, were extracted from the content logs and transcripts.

7.3.3 Findings

7.3.3.1 Number of clicks

The number of clicks used to complete each task was retrieved from the recorded videos to measure the ease of navigation for each component. These results were averaged to find the average number of clicks for each task and compared to the minimum number of clicks possible to complete this task. Figure 7.4 displays a graph comparing the minimum number of clicks to the average number of clicks.

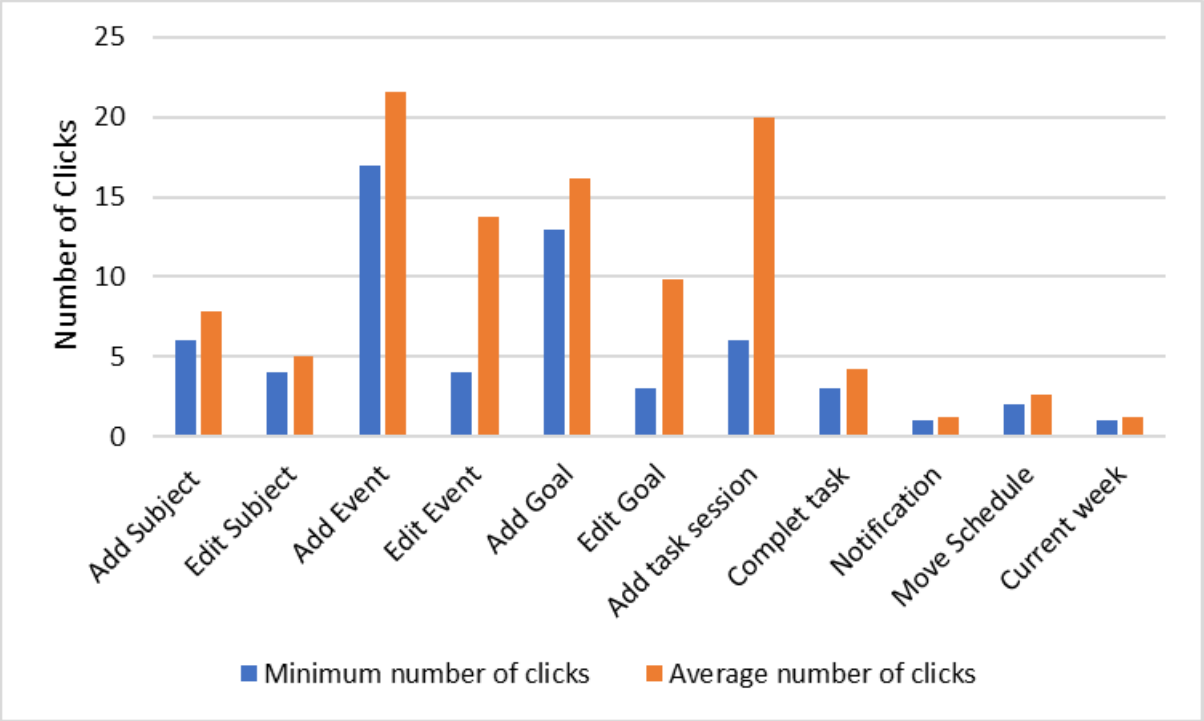


Figure 7.4: Chart displaying the comparison between the minimum number of clicks and the average number of clicks per task

7.3.3.2 Error rate per task

The usability test documented errors for each task by reviewing the recorded videos. Errors included incorrect inputs, misusing features, and navigation issues. However, only navigation issues were prevalent enough to yield significant findings. Consequently, navigation errors were the primary focus when recording issues for each task. To calculate the error rate, the total number of navigation errors made by all participants for a specific task was divided by the total number of attempts for that task. The result was then multiplied by 100 to express the error rate as a percentage. Figure 7.5 displays the error rate per task.

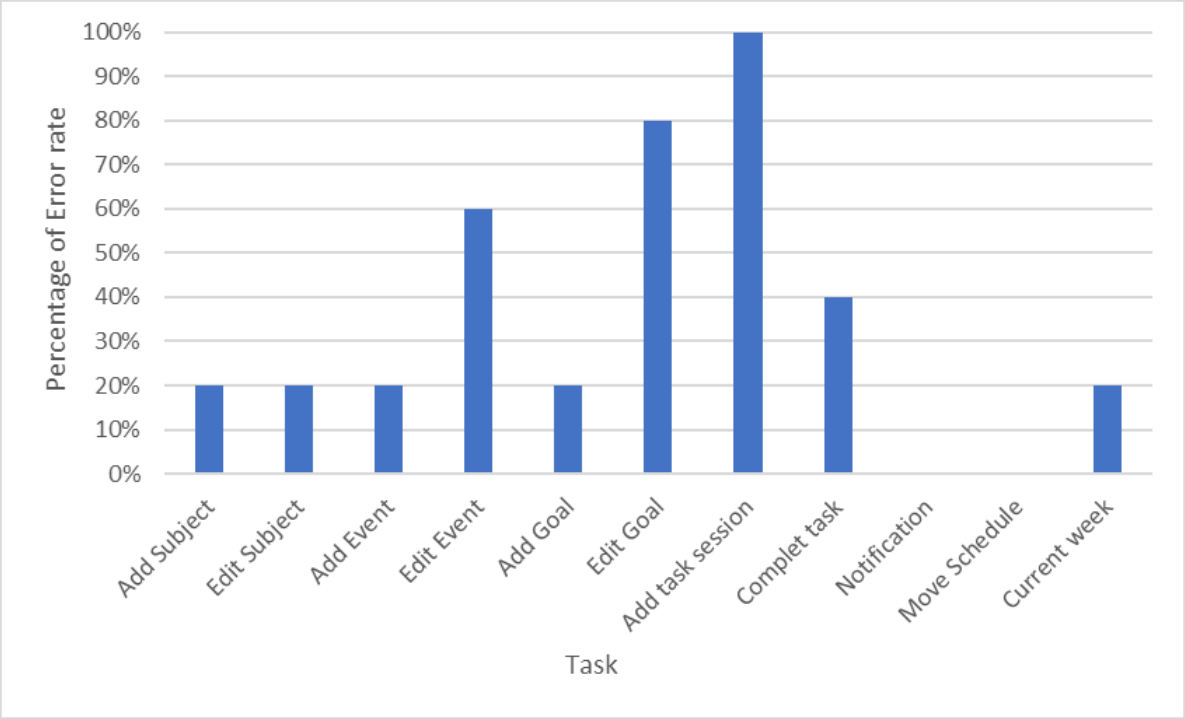


Figure 7.5: Chart displaying the error rate per task

7.3.3.3 Interviews

Once the interviews were completed, the data was transcribed. Following this, themes and patterns of the data were discovered. The following section will detail a summary of responses for each question. (The full transcript is in appendix 1)

Question 1 - Did you find it easy setting up a schedule of events in the app? What did you find easy or difficult with this task?

- Easy event creation: 5/5 participants found creating events straightforward.
- Difficult task session: 4/5 participants reported difficulty in understanding how to add a task session.

Question 2 - Were there any features or options you felt were missing or could be improved when creating a schedule of events?

- Error handling: 1/5 participants mentioned the absence of error handling.
- Editing and deleting events: 3/5 participants suggested clearer indications for editing events and better interaction with buttons for deleting tasks.

Question 3 - Did you find it easy setting up goals in the app? What did you find easy or difficult with this task?

- Easy goal setting: 5/5 participants found goal setting to be easy.

Question 4 - Were there any features or options you felt were missing or could be improved when creating a goal?

- Visual indicators: 2/5 participants suggested adding visual cues for completed goals and changing the progress bar color.
- Additional features: 2/5 participants suggested adding rewards for goal completion and improving the task deletion buttons.

Question 5 - How well do you think the app integrates the scheduling of events and goal-setting features?

- Positive integration: 4/5 participants thought the app integrated both features well.
- Task session improvement: 4/5 participants highlighted the need for better task session functionality to fully understand the integration

Question 6 - Do you think the notifications feature is useful?

- Generally useful: 5/5 participants found the notifications feature useful.
- Customisation and control: 1/5 participants expressed concerns about potential annoyance and desired more control over notifications.

Question 7 - Overall, how satisfied are you with the app's features?

- 3/5 participants were satisfied
- 2/5 participants were somewhat satisfied

Question 8 - Do you think the app would be useful in terms of helping you with time management related to your studies? Would you use it if it was available? Please explain your answer.

- Useful for time management: 5/5 participants agreed that the app could be helpful for managing their time.
- Usage and duration: 4/5 participants would use the app if available, but 2/5 participants expressed uncertainty about how long they would continue using it.

Question 9 - How likely would you be to recommend this app to a friend or fellow student?

- Recommendation: 4/5 participants would recommend the app to friends or fellow students.
- Task session concerns: 3/5 participants were hesitant to recommend the app due to the issues with the task session functionality.

7.3.4 Discussion

The following section reflects on the Findings from the usability testing and demonstrates how the results were used to improve the requirements and the application's design.

The usability test data reveals that editing an event is a cumbersome process, as evidenced by the fact that it requires over three times the number of clicks shown in Figure 7.4. Moreover, the edit event functionality exhibits a substantial 60% error rate, as depicted in Figure 7.5. Furthermore, three out of five participants suggested enhancing the clarity of the event editing process.

This feedback suggests that users may face difficulty in discerning the correct approach to editing an event. Presently, users need to click the calendar box to edit a goal, which deviates from the editing process for goals and subjects, where a pencil icon is provided. The event display was updated to address this issue, as illustrated in Figure 6.4. The revised event display, shown in Figure 6.5, now incorporates an edit pencil icon, signifying the method for editing an event. Utilising a pencil icon ensures consistency with other editing buttons throughout the application.

Figure 6.4: Original event display

Figure 6.5: Redesigned event display

Figure 7.4 illustrates that adding a goal required an average of just over seven clicks, which is unusually high for such a straightforward task. Additionally, Figure 7.5 reveals that 80% of users encountered navigation difficulties when attempting to edit a goal. In the post-test interviews, three out of five participants expressed needing clearer indications for deleting tasks.

It can be suggested that the primary reason for the high error rate is the lack of clarity regarding task deletion. To address this issue, the design of the edit goal form has been revised to enhance the visibility and understanding of task deletion. As displayed in Figure 6.5, the updated task section of the goal form now incorporates a trash bin icon, which more effectively conveys the method for removing tasks to the users. This improvement aims to reduce the error rate and streamline the user experience.

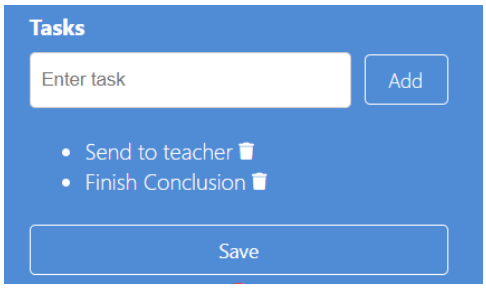


Figure 6.6: Redesigned task management section

Task selection posed the most significant challenges, as adding a task session became the most error-prone process, requiring an average of 16 extra clicks, as illustrated in Figure 7.4. All users experienced navigation difficulties with this component, as depicted in Figure 7.5, and 4 out of 5 users mentioned this issue during their interviews. This problem led 3 out of 5 participants to hesitate to recommend the app, emphasising the need for a solution. It can therefore be suggested that the application lacks details to allow the user to understand how to create a task session.

In response to the feedback, a toolbar button was introduced, which, when clicked, displayed a 5-second GIF demonstrating how to add a task session. Unfortunately, the screen recorders used to create the GIF failed to capture the dropdown options accurately, leaving the issue unresolved due to time constraints. Figure 6.7 showcases the demonstration popup.

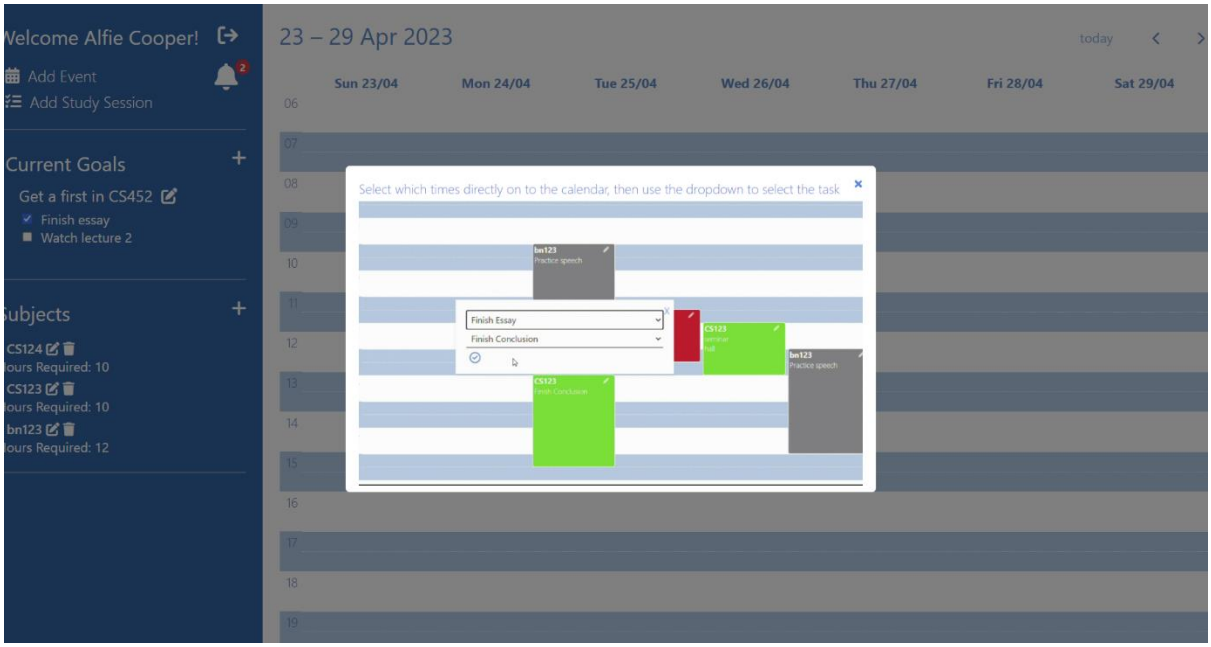


Figure 6.7: Popup displaying task session creation instructions

Despite time constraints, essential design and requirement modifications were implemented based on the findings from the usability testing. However, it was not feasible to conduct another round of usability testing to evaluate the effectiveness of these changes.

7.4 Conclusion

In conclusion, this chapter has provided a comprehensive overview of the testing strategies employed during the application development, including automated testing, manual testing, and usability testing. Despite the limitations imposed by time constraints, the testing phase proved to be a crucial step in the development process, revealing areas that required improvement. The feedback gathered through usability testing led to design changes that enhanced the application's usability and user experience.

One significant aspect not addressed during the testing phase was the application's long-term usability. As noted in the literature review, students may struggle to maintain motivation for using

time management techniques, such as scheduling, over extended periods (Wajcman, 2018). To accurately assess the true usability of this application, it would be necessary to evaluate a student's usage over a longer duration. Unfortunately, this was not possible within the project's limited time frame. Nonetheless, the usability testing contributed to substantial improvements in the application and brought attention to issues not recognised during the design and development stages.

8 Evaluation

This evaluation chapter aims to assess the project's success in meeting the specified functional and non-functional requirements, providing a comprehensive understanding of its overall effectiveness. This chapter will evaluate each requirement in detail and discuss how well the project has achieved its goals. Moreover, it will touch upon future development opportunities and areas for improvement while also reflecting on the development process, challenges faced, and lessons learned.

This project aims to create an application for students to manage better and prioritise their time. Evaluating a project against its requirements is critical in ensuring that it meets stakeholder expectations and delivers on its objectives. Evaluating against the requirements gives a better understanding of the project's success. The following section evaluates whether the project has matched the functional and non-functional requirements.

8.1 Functional requirements

8.1.1 Requirements 1 & 2 – System will allow the user to login & System will allow the user to create-an-account

All authentication processes were successfully completed in evaluating the user authentication requirement, ensuring that only authorised users can access the system. Although usability testing did not include user authentication due to ethical concerns surrounding the use of personal credentials, manual testing was employed to assess the effectiveness of these requirements. As demonstrated in Table 6.1, the manual testing results confirmed that the user authentication requirement has been met, providing a secure environment for users and maintaining the system's integrity.

8.1.2 Requirement 3 - The system will have a schedule for each user

Requirement 3.1 was deemed successful, as 100% of users in the usability test found event creation easy. Furthermore, manual and usability testing confirmed that requirements 3.2-3.6 were effectively implemented and functioning as intended. However, sections 7.3.3.1 and 7.3.3.2 highlight user difficulties locating the edit event feature, which was subsequently addressed, as detailed in section 7.3.4. Due to time constraints, requirement 3.7 could not be implemented. Nevertheless, since requirement 3.7 was labelled as a "Could Have" feature, its absence is considered insignificant. Thus, the overall requirement has been satisfactorily met despite the unimplemented feature.

8.1.3 Requirement 4 – The system will allow the user to set goals

Requirement 4 was successfully implemented, with 80% of usability test participants indicating that goal setting was effectively incorporated into the system. Moreover, all five participants confirmed that requirement 4.1 was successfully implemented. Although usability testing initially revealed an issue with requirement 4.2, the section was redesigned, as demonstrated in Figure 6.6, to address the identified concerns and ensure a seamless user experience for goal setting.

8.1.4 Requirement 5 – The system should allow the user to enter their subjects

The evaluation of requirement 5, based on findings from both manual testing and usability testing, suggests that it was successfully met. For instance, usability test results demonstrate that all users could add and delete subjects with minimal error rates, as shown in Figure 7.3.3.2. This evidence indicates that the system effectively supports adding and deleting subjects, meeting the specified requirement and contributing to a positive user experience.

8.1.5 Requirement 6 – The system should give suggestions on how to improve their time management.

Although the requirement was initially poorly defined, it has been considered met due to the successful implementation of extensive features. These features include the language model, which provided solutions for users' time management inputs, and the data analysis functions that responded to specific criteria, such as identifying instances where lower priority goals had more time allocated than higher priority goals or when time spent on each subject was unbalanced. Requirement testing, conducted through both manual and automated methods, proved successful. Manual testing demonstrated that the components worked cohesively and the outputs were displayed accurately, while unit testing confirmed the proper operation of the data analysis functions, as illustrated in Figure 7.3. Overall this requirement was a success as 5/5 participants found the notifications useful within the usability test

8.2 Non-functional requirements

8.2.1 Usability

The usability test results indicate that the application has successfully met the requirement, with 100% of participants stating that the app is useful for time management. Additionally, 80% of the responses expressed a willingness to use the app if available and to recommend it to a friend. Although the usability test identified some issues related to the usability of certain components, design changes detailed in section 7.3.4 have addressed most of these concerns, further demonstrating the application's effectiveness in meeting the specified requirement.

8.2.2 Security

Security measures such as SQL Injection prevention and password encryption were successfully implemented. However, aspects of this application may still be vulnerable as restricted development time limited the protections implemented.

8.2.3 Compatibility

A web application was selected as the chosen application type to increase the compatibility across devices, as discussed in section 5.1.1. The application was tested using manual testing by running the application on different browsers and devices.

8.3 Future development

In developing this application, it was crucial to incorporate extensive features to distinguish it within the saturated time management app market. However, the short development time impacted the overall quality of the application, despite the successful implementation of these features. As such, future development priorities include delivering a more polished application with enhanced security features, such as protection against cross-site forgery and improved authentication session management. Additionally, completing all requirements, such as requirement 3.7, would be addressed in future development efforts. Further usability testing will be conducted to assess the impact of the changes made in section 7.3.4. Moreover, additional background research will be carried out to identify features that cater to specific demographics, addressing the gap in section 2.1.6.

8.4 Reflection

This chapter has been written in first person format as it accounts for my personal reflection. In this reflection chapter, I share my thoughts on the development process of the web application I recently completed. Although the final deliverable is a functional web application that meets all the requirements, there were certain disappointments and challenges I faced during the project, as well as areas where I was able to improve my professional and technical competencies.

As mentioned in section 8.3, creating a product with unique and extensive features was prioritised over overall quality of the application as it was important to distinguish this deliverable upon a saturated time management market. Therefore one of the most significant disappointments I experienced was the little time for developing in a more polished product. This means that the application's input validation is lacking, which can potentially cause errors throughout the system. Additionally, I would have liked to research more about front-end design to create a better user interface, but unfortunately, I lacked experience in that area. Moreover, testing frameworks could have been implemented more successfully, and more information could have been retrieved, especially for usability testing.

Despite these setbacks, the development process allowed me to improve certain professional and technical competencies. The background research stage helped me enhance my critical thinking and analysis skills when reviewing professional papers. Furthermore, changes in the design during the development stage pushed me to improve my adaptability and organisation skills. Additionally, I believe that my skills in web-based technologies have grown significantly, ranging from comprehending how web servers interact with external services to gaining a deeper understanding of front-end frameworks.

On a positive note, I am very pleased with the overall outcome of the project. The delivered web application is fully functional and usable, and it provides features such as the language model and data analysis that were not reviewed within the app analysis. The system still needs minor improvements to provide a more polished app to market, but I believe it has a solid unique selling proposition (USP) that the public would desire.

9 Bibliography

Aayish, Z., n.d. Session Cookies in Node.js. Geeks for Geeks. Available at: https://www.geeksforgeeks.org/session-cookies-in-node-js/ [Accessed 8 April 2023].

Dutta, S., 2020. How to Authenticate Users in Your Node App with Cookies, Sessions, and Passport.js. FreeCodeCamp. Available at: https://www.freecodecamp.org/news/authenticate-users-node-app/ [Accessed 1 April 2023].

Quick, J.Q., 2022. How to prevent SQL injection attacks in Node.js. Planetscale. Available at: https://planetscale.com/blog/how-to-prevent-sql-injection-attacks-in-node-js [Accessed 12 April 2023].

Kolodiy, M., n.d. Developers. Unit Testing and Coding: Why Testable Code Matters. Toptal. Available at: https://www.toptal.com/qa/how-to-write-testable-code-and-why-it-matters [Accessed 15 April 2023].

Lee, B., 2017. Quick Dig: util.promisify in Node.js. Medium. Available at: https://codeburst.io/quick-dig-promisify-in-node-js-6d5d763f847d [Accessed 10 April 2023].

Mecinski, M., 2019. Node.js, MySQL and async/await. Medium. Available at: https://codeburst.io/node-js-mysql-and-async-await-6fb25b01b628 [Accessed 3 April 2023].

Mordecai, N., 2019. How to Implement Multiple Queries Transactions with Node.JS & MySQLWriting MySQL transactions in NodeJS & TypeScript. Nicholasmordecai.co.uk. Available at: https://nicholasmordecai.co.uk/database/transactions-with-multiple-queries-nodejs-mysql/ [Accessed 7 April 2023].

WebDev Simplified, 2020. Node.js Passport Login System Tutorial. Youtube. Available at: youtube.com/watch?v=-RCnNyD0L-s&list=PLZlA0Gpn_vH9yI1hwDVzWqu5sAfajcsBQ [Accessed 31 March 2023].

10 References

Adams, R. V., & Blair, E., 2019. Impact of Time Management Behaviors on Undergraduate Engineering Students' Performance. SAGE Open, 9(1). https://doi.org/10.1177/2158244018824506. Accessed October 27, 2022.

Albar, R., Mohamed, A., Albarazi, M., McAleer, S., & Sharibah, H. (2022). Interplay between personality traits and learning strategies: the missing link. Accessed December 2, 2022.

Ariely, D., & Wertenbroch, K., 2002. Procrastination, deadlines, and performance: Self-control by precommitment. Psychological Science, 13(3), pp. 219-224. Accessed November 29, 2022.

Bahrami, Z., Heidari, A., & Cranney, J. (2022). Applying SMART Goal Intervention Leads to Greater Goal Attainment, Need Satisfaction and Positive Affect. International Journal of Mental Health Promotion, 24(6), 869-882. Accessed November 15, 2022.

Beattie, G., Laliberté, J. P., Michaud-Leclerc, C., & Oreopoulos, P., 2017. What sets college thrivers and divers apart? A contrast in study habits, attitudes, and mental health. National Bureau of Economic Research Working Paper No. 23588. Accessed December 4, 2022.

bcrypt, 2021. Bcrypt password hashing library. Retrieved April 18, 2023, from https://github.com/kelektiv/node.bcrypt.js

body-parser, 2020. Node.js body parsing middleware. Retrieved April 17, 2023, from https://github.com/expressjs/body-parser

Britton, B. K., & Tesser, A., 1991. Effects of time-management practices on college grades. Journal of Educational Psychology, 83, 405–410. Accessed November 10, 2022.

Busato, V. V., Prins, F. J., Elshout, J. J., & Hamaker, C. (2000). Intellectual ability, learning style, personality, achievement motivation and academic success of psychology students in higher education. Personality and Individual Differences, 29, 1057–1068. Accessed November 5, 2022.

Cirillo F, 2013. The Pomodoro® Technique. https://francescocirillo.com/products/the-pomodoro-technique. Accessed November 20, 2022.

Clegg, D. (2013). Managing Successful Projects with PRINCE2 2017 Edition. TSO (The Stationery Office). Retrieved April 15, 2023, from https://www.axelos.com/Corporate/media/Files/Sample_Papers/PRINCE2_2017_Foundation_Sample_Paper_1.pdf

Covington, M. V., 2000. Goal theory, motivation, and school achievement: An integrative review. Annual Review of Psychology, 51, 171–200. https://doi.org/10.1146/annurev.psych.51.1.171. Accessed October 28, 2022.

cookie-parser, 2021. Node.js cookie parsing middleware. Retrieved April 19, 2023, from https://github.com/expressjs/cookie-parser

DayPilot, 2023. JavaScript event calendar and scheduling components. Retrieved April 19, 2023, from https://www.daypilot.org/

EJS (2023). Embedded JavaScript templating. Retrieved April 18, 2023, from https://ejs.co/

Fang Liu, Yanan Xu, Tianshuai Yang, Zhihua Li, Yakun Dong, Liang Chen, Xiaohua Sun, 2022. The Mediating Roles of Time Management and Learning Strategic Approach in the Relationship Between smartphone Addiction and Academic Procrastination. Psychology Research and Behavior Management, Volume 15, pp. 2639 – 2648. Accessed December 2, 2022.

Fielding, R.T. and Taylor, R.N., 2002. Principled design of the modern Web architecture. ACM Transactions on Internet Technology (TOIT), 2(2), pp.115-150. Retrieved April 15, 2023.

Font Awesome, 2023. Iconic font and CSS toolkit. Retrieved April 20, 2023, from https://fontawesome.com/

FullCalendar, 2023. Full-featured JavaScript event calendar. Retrieved April 16, 2023, from https://fullcalendar.io/

Gobbo, F., Vaccari, M., 2008. The Pomodoro Technique for Sustainable Pace in Extreme Programming Teams, Agile Processes in Software Engineering and Extreme Programming. Accessed November 3, 2022.

Gray, E. K., & Watson, D. (2002). General and specific traits of personality and their relation to sleep and academic performance. Journal of Personality, 70, 177-206. Accessed October 29, 2022.

Guàrdia, L., Maina, M., & Sangrà, A., 2013. MOOC design principles. A pedagogical approach from the learner's perspective. eLearning Papers, 33, 1–6. Accessed November 15, 2022.

Hanushek, E.A. & Woessmann, L., 2006. The Role of Education in Economic Growth and Development: A Theoretical and Empirical Analysis. In: E.A. Hanushek & F. Welch, eds. Handbook of the Economics of Education, Volume 1. Amsterdam: Elsevier, pp. 67-200. Accessed November 1, 2022

IBM (2021). PostgreSQL vs. MySQL: What's the difference? IBM Cloud Blog. Retrieved April 16, 2023, from https://www.ibm.com/cloud/blog/postgresql-vs-mysql-whats-the-difference

Jinalee & Singh. (2018). A descriptive study of time management models and theories. International Journal of Advanced Scientific Research and Management, 3(9). Accessed November 20, 2022.

Kaner, C., Falk, J., & Nguyen, H. Q. (1999). Testing Computer Software. New York: Wiley. Retrieved April 20, 2023.

Karas, D., Marcantonio, M.S., & Spada, M., 2009. Brief behavioral-cognitive coaching for procrastination: A case series. Coaching: An International Journal of Theory Research and Practice, 2(1), pp. 44-53. Accessed November 11, 2022.

Kember, D., Leung, D.Y.P., McNaught, C., 2008. A workshop activity to demonstrate that approaches to learning are influenced by the teaching and learning environment. Active Learning in Higher Education, 9, pp. 43–56. Accessed November 18, 2022.

Kyung Ryung Kim, Eun Hee Seo, 2015. The relationship between procrastination and academic performance: A meta-analysis. Personality and Individual Differences, Volume 82, Pages 26-33. Accessed November 6, 2022.

Lay, C.H., 1986. At last, my research article on procrastination. Journal of Research in Personality, 20, pp. 474-495. Accessed November 4, 2022.

LeCun, Y., Bengio, Y., & Hinton, G., 2015. Deep learning. Nature, 521(7553), pp. 436–444. Accessed November 9, 2022.

LinkedIn Learning, 2023. Online courses and video tutorials. Retrieved April 21, 2023, from https://www.linkedin.com/learning/

Material-UI, 2023. React components for faster and easier web development. Retrieved April 17, 2023, from https://material-ui.com/

McCown, W., Petzel, T., & Rupert, P., 1987. An experimental study of some hypothesized behaviors and personality variables of college student procrastinators. Personality and Individual Differences, Volume 8, Issue 6, pp. 781-786. Accessed October 31, 2022.

Microsoft, 2023. Microsoft Teams. Retrieved April 22, 2023, from https://www.microsoft.com/en-us/microsoft-teams/group-chat-software

McKenzie, K., & Gow, K. (2004). Exploring the first-year academic achievement of school leavers and mature-age students through structural equation modelling. Learning and Individual Differences, 14(2), 107-123. Accessed November 5, 2022

Moodle, 2023. Open-source learning platform. Retrieved April 23, 2023, from https://moodle.org/

Murray, N., Unsworth, C., & Hanley, P., 2011. The resource-based learning engagement: A study of the learning interactions between a student and a set of resources. Educational Review, 63, pp. 151–166. Accessed November 21, 2022.

Node.js, 2023. JavaScript runtime built on Chrome's V8 JavaScript engine. Retrieved April 18, 2023, from https://nodejs.org/en/

Nordgren, L.F., van der Pligt, J., & van Harreveld, F., 2008. The instability of health cognitions: Visceral states influence self-efficacy and related health beliefs. Health Psychology, 27(6), pp. 722-727. Accessed November 27, 2022.

Oracle, 2023. MySQL: The world's most popular open source database. Retrieved April 16, 2023, from https://www.mysql.com/

Ozer, B. U., Demir, A., & Ferrari, J. R. (2009). Exploring academic procrastination among Turkish students: Possible gender differences in prevalence and reasons. The Journal of Social Psychology, 149(2), 241-257. Accessed November 6, 2022

PostgreSQL, 2023. The world's most advanced open source relational database. Retrieved April 17, 2023, from https://www.postgresql.org/

React, 2023. A JavaScript library for building user interfaces. Retrieved April 15, 2023, from https://reactjs.org/

Schraw, G., Wadkins, T., & Olafson, L., 2007. Doing the things we do: A grounded theory of academic procrastination. Journal of Educational Psychology, 99(1), pp. 12-25. Accessed October 28, 2022.

Semb, G., Glick, D.M., & Spencer, R.E., 1979. Student withdrawals and delayed work patterns in self-paced psychology courses. Teaching of Psychology, 6(1), pp. 23-25. Accessed November 5, 2022.

Semb, G., Glick, D.M., & Spencer, R.E., 1979. Student withdrawals and delayed work patterns in self-paced psychology courses. Teaching of Psychology, 6(1), pp. 23-25. Accessed November 5, 2022.

Solomon, L.J., & Rothblum, E.D., 1984. Academic procrastination: Frequency and cognitive-behavioral correlates. Journal of Counseling Psychology, 31(4), pp. 503-509. Accessed November 2, 2022.

Steel, P. (2007). The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure. Psychological Bulletin, 133(1), 65-94. Accessed November 2, 2022

Trueman, M., & Hartley, J. (1996). A comparison between the time-management skills and academic performance of mature and traditional-entry university students. Higher Education, 32(2), 199-215.Accessed November 7, 2022.

Tuckman, B.W., 1991. The development and concurrent validity of the procrastination scale. Educational and Psychological Measurement, 51(2), pp. 473-480. Accessed November 7, 2022.

Tuckman, B.W., & Schouwenburg, H.C., 2004. Procrastination, performance, and stress. In H.C. Schouwenburg, C.H. Lay, T.A. Pychyl, & J.R. Ferrari (Eds.), Counseling the procrastinator in academic settings (pp. 61-73). Washington, DC: American Psychological Association. Accessed November 3, 2022.

van Eerde, W., 2003. A meta-analytically derived nomological network of procrastination. Personality and Individual Differences, 35(6), pp. 1401-1418. Accessed November 1, 2022.

Visual Studio Code, 2023. Code editing redefined and optimized for building and debugging modern web and cloud applications. Retrieved April 20, 2023, from https://code.visualstudio.com/

Wolters, C.A., 2003. Understanding procrastination from a self-regulated learning perspective. Journal of Educational Psychology, 95(1), pp. 179-187. Accessed October 30, 2022.

Zimmerman, B.J., & Kitsantas, A., 2005. Homework practices and academic achievement: The mediating role of self-efficacy and perceived responsibility beliefs. Contemporary Educational Psychology, 30(4), pp. 397-417. Accessed November 8, 2022.

Zimmerman, B.J., & Schunk, D.H., 2011. Self-regulated learning and performance: An introduction and an overview. In B.J. Zimmerman & D.H. Schunk (Eds.), Handbook of self-regulation of learning and performance (pp. 1-12). New York, NY: Routledge. Accessed November 10, 2022.

Appendix

1 Usability test transcript

1. 1 Participant 1

Did you find it easy setting up a schedule of events in the app? What did you find easy or difficult with this task?

Setting up events were straightforward, nothing was particularly difficult however I thought the cancel button was in the incorrect place

Were there any features or options you felt were missing or could be improved when creating a schedule of events?

You should check the inputs to make sure they are correct as I think I submitted the wrong value and I couldn't check

Did you find it easy setting up goals in the app? What did you find easy or difficult with this task?

Setting up the goals was straightforward as well, I would have assumed that there was hover help however, there was none.

Were there any features or options you felt were missing or could be improved when creating a goal?

I would have liked to have hover help so as so I would know what each input was requiring more clearly. Also I thought the buttons to delete tasks were unclear as the just looked like 'X''s which made them look like apart of the task text

How well do you think the app integrates the scheduling of events and goal-setting features?

Yes I thought this was okay if you knew what you are doing however it would be useful to have more guidance especially the feature to add task sessions.

Do you think the notifications feature is useful?

Yes I thought this was essential

Overall, how satisfied are you with the app's features?

I am satisfied

Do you think the app would be useful in terms of helping you with time management related to your studies? Would you use it if it was available? Please explain your answer.

Yes I think it would and I would use it because it gives me a good overall indication on how I am progressing with my tasks.

How likely would you be to recommend this app to a friend or fellow student?

I am likely to recommend this to another student

1.2 Participant 2

Did you find it easy setting up a schedule of events in the app? What did you find easy or difficult with this task?

Creating an event was easy and I liked how you could see the colour of the subject on the calendar however I did think the crosses next to the event type were unnecessary

Were there any features or options you felt were missing or could be improved when creating a schedule of events?

I would remove this crosses as previously mentioned.

Did you find it easy setting up goals in the app? What did you find easy or difficult with this task?

I was able to create goals easily however I found the display for the select menus confusing as they looked like text boxes

Were there any features or options you felt were missing or could be improved when creating a goal?

I thought the task deletion buttons were confusing

How well do you think the app integrates the scheduling of events and goal-setting features?

It was all good however, I found it difficult to know what to do to add a task session

Do you think the notifications feature is useful?

Yes I could see how it would improve how I varied out my time

Overall, how satisfied are you with the app's features?

Somewhat satisfied

Do you think the app would be useful in terms of helping you with time management related to your studies? Would you use it if it was available? Please explain your answer.

Yes I think this is a better alternative for normal schedules as it allows me to set goals, however the task session feature must improve

How likely would you be to recommend this app to a friend or fellow student?

I would definitely recommend this to a friend as I know a lot of them have bad time management

1.3 Participant 3

Did you find it easy setting up a schedule of events in the app? What did you find easy or difficult with this task?

Adding events was easy, however I could not figure out how to had a task session

Were there any features or options you felt were missing or could be improved when creating a schedule of events?

An improvement would be to add instructions on how to add a task session as it is impossible to know otherwise

Did you find it easy setting up goals in the app? What did you find easy or difficult with this task?

Yes goal setting was very easy, there were no difficulties

Were there any features or options you felt were missing or could be improved when creating a goal?

It would help making the new goal button a bit more clear, it is a bit small compared to the other buttons. I also think it would be good to add an indication on when the goal is past its due date

How well do you think the app integrates the scheduling of events and goal-setting features?

Apart from adding task session I think the integration works well, I like how you can add subjects to goals and events

Do you think the notifications feature is useful?

I think it would help initially but may get annoying after time, I also didn't know what the search bar was for

Overall, how satisfied are you with the app's features?

Somewhat satisfied

Do you think the app would be useful in terms of helping you with time management related to your studies? Would you use it if it was available? Please explain your answer.

Yes I think I would use this however im not sure how long I would use it for as I have used similar apps but have never stuck with them

How likely would you be to recommend this app to a friend or fellow student?

I probably wouldn't recommend it at this stage due the the task session not working

1.4 Participant 4

Did you find it easy setting up a schedule of events in the app? What did you find easy or difficult with this task?

The actual event was easy to add. It was impossible to know how to add task session

Were there any features or options you felt were missing or could be improved when creating a schedule of events?

I thought it was difficult to know how to edit the task session or event. Maybe add an indication on that pressing them would edit them, changing how the mouse interacts with the buttons will help to determine what is clickable and what is not

Did you find it easy setting up goals in the app? What did you find easy or difficult with this task?

Yes goals were easy. However adding tasks were not that clear

Were there any features or options you felt were missing or could be improved when creating a goal?

No not really apart from adding tasks

How well do you think the app integrates the scheduling of events and goal-setting features?

I think the app integrated both features well if there was a better indication of how to add task sessions

Do you think the notifications feature is useful?

I couldn't use the app enough to really find it useful however I think it would be useful once using it for a longer period of time

Overall, how satisfied are you with the app's features?

Somewhat satisfied

Do you think the app would be useful in terms of helping you with time management related to your studies? Would you use it if it was available? Please explain your answer.

Yes I think I would use this however im not sure how long I would use it for

How likely would you be to recommend this app to a friend or fellow student?

Yes I would recommend this as I haven't seen anything similar before.

1.5 Participant 5

Did you find it easy setting up a schedule of events in the app? What did you find easy or difficult with this task?

It was easy to add events but unclear how to add task sessions

Were there any features or options you felt were missing or could be improved when creating a schedule of events?

It would be good to Include an add task to board demonstrating how to add it

Did you find it easy setting up goals in the app? What did you find easy or difficult with this task?

Yes adding goals was easy

Were there any features or options you felt were missing or could be improved when creating a goal?

I think it would be good to view the remaining days. Also I think it would be good if there were rewards to give me an incentive

How well do you think the app integrates the scheduling of events and goal-setting features?

Very well I thought it had a really good layout

Do you think the notifications feature is useful?

yes

Overall, how satisfied are you with the app's features?

Very satisfied

Do you think the app would be useful in terms of helping you with time management related to your studies? Would you use it if it was available? Please explain your answer.

Yes, I like thatit makes you split goals into taks and add it into calendar to visualize your time and shows the progress you are making I also like how little effort you get a lot of information

How likely would you be to recommend this app to a friend or fellow student?

Very likely

2 Questionnaire plan

- What is your age
- What is your highest level of education?
- What subject do/did you study?
- How do/did you typically manage your time in terms of your academic studies?
- What do/did you find most challenging in terms of managing your time for your studies?
- When studying, do/did you find yourself procrastinating (i.e., putting of attending to your studies)?
    - If applicable, what are/were your main causes of procrastination?
- When studying, do/did you find that you can get distracted?
    - If applicable, what are/were you most often distracted by?
- Do/did any of the following negatively impact your studies? *(tick all that apply)*
    - Lack of motivation
    - Failing to prioritize tasks
    - Bad time estimation
- Do/did you currently use any time management techniques to help you study? Yes/No
    - If yes:
        - What techniques? Please describe them.
        - What aspects of these techniques do/did you find useful and why?
    - If no, why not?
- Have you used any other time management techniques in the past? Yes/No
    - If yes:
        - What techniques? Please describe them.
        - Why did you stop using them?
- Do/did you currently use any time management apps/software/websites? Yes/No
    - If yes:
        - What apps/software/websites do/did you use?
        - What features of these platforms do/did you find useful?
        - What features would you like to be included?
    - If no, why not?
- Have you used any different time management apps/software/websites in the past?
    - If yes:
        - What apps/software/websites did you use?
        - What features of these platforms did you find useful?
        - What was the reason for you stopping using these?