UNIVERSITY OF LONDON

BSc Computer Science

CM3070 FINAL PROJECT
FINAL PROJECT DRAFT REPORT

Task Manager Mobile Application

Name: Shanath S/O Rajendran
UoL ID: 210470552

Contents

Introduction

The template chosen for this final project, is the Project Idea Title 1: Task Manager Mobile App from CM3050 Mobile Development.

The idea for this project comes from my own personal experience, where I have downloaded task manager applications in the past, but never consistently used them because the application was either not engaging enough, or it was aesthetically unappealing.

Why is this important?

We live in a fast-paced world, most of us reside in cities. City life is stressful. We have lots of responsibility and obligations to fulfil on a regular basis. We might have multiple places to be in a day. Thus, we need a way to track and manage our tasks. Coupled with the fact that we have too many distractions in the modern world, due to rapidly emerging technology, like social media and games, it can be hard to stay focused on our responsibilities.

The objectives for this final project are as follows:

- Define the scope of the project
- Conduct a literature review to support the development of the project
- Build a task manager mobile application for Android
- Allow users to take notes, create a checklist with calendar integration, utilize a timer for time management, and project tool to undertake large scale tasks.
- Deploy it to the Google Play Store
- Have my target demographic review the application, and ascertain if it outperforms what's currently out on the market.

The deliverables for this final project are as follows:

- An initial preliminary report showcasing the chosen project template and the path ahead.
- The source code for the mobile application.
- A final report detailing the entirety of the project, including the full development, application testing, user feedback compilation and deployment unto the Google Play Store.
- A video demonstration showcasing the mobile application at work.
- The evaluation results of the project.

The following sections will comprise of literature reviews to support the need of having a time management and project management tool. Followed by an overview of the intended audience, technologies to be used, wireframes that will be tested by survey respondents, and a Gantt Chart outlining the path ahead in completing this final project, finishing off with a prototype implementation of the discussed features.

Literature Review

In today's fast-paced, digitally distracting world, effective time management and goal-setting are crucial for students and adults. This literature review informs the development of a task manager application by exploring three key areas: **Time Management Techniques**, **Goal Setting and Visualization** and **Comparative Analysis of Productivity Apps**.

Time Management

Nasrullah and Khan from the University of Peshawar and Gomal[1], conducted a study to evaluate the strengths and weaknesses of time management practices on student academic performance. Recognizing the importance of proper time management in today's competitive environment[2], the study focused on time management mechanisms and goal setting among students from Qurtuba University of Science and Technology, Pakistan. Surveys assessed students' short-term and long-term time management capabilities. The data analysis included descriptive statistics and correlation analysis to explore the relationship between time management practices and academic performance.

The results showed a positive correlation between time management and the student’s academic performance. It revealed that there is a connection when it comes to academic performance and stress reduction, when utilizing time management practices. The study showed relevance and practicality to address the issue of student’s academic performance as well as using robust analytical tools to tabulate their findings. However, the sample size chosen was rather small, only a total of 120 students, lacking diversity of participants form various regions. Therefore, care has to be taken as it may not be wise to generalize the findings to all student populations globally.

A research conducted by Albulescu and colleagues, from the University of Timioara, Romania[3], delved into the effectiveness of taking micro-breaks and its impact on performance. The study was an analysis of conducted studies in the relevant field over the last 30 years on the subject of micro breaks, on whether taking short breaks of 10 minutes or less, when performing a task, leads to better performance without accumulating too much strain on the person to the point where it affects their health.

Out of 4868 case studies drafted, 22 studies were finalized and narrowed down for the analysis. The participants chosen in the selected studies were healthy individuals which included both students and young adults. The measurements that were monitored and classified in this analysis were vigor, fatigue or performance. Vigor being an individual’s inclination to carry on with the task at hand, even when challenges present themselves, while fatigue relates to how tired the individual is. The performance measurements relate to whether the individual was able to perform his or her task with high degree of accuracy and precision. Tasks participants had to perform were classified into three categories. Creative, clerical and cognitive tasks. All 22 studies had a control group as a comparison, such as those with micro break and those without.

The collective findings of the chosen studies concluded that micro breaks do indeed improve the performance and vigor of the individual whilst reducing fatigue for clerical and creative tasks, though there was barely any improvement for cognitive intensive tasks. The data uncovered that those who took breaks that were longer than 10 minutes, generally had better performance.

However, there were several key factors that had to be considered for further study. The duration of the micro break could not be decisively agreed upon by experts in the field. The tasks in which an individual partakes prior to the break, also mattered greatly, as well as the activity the individual partakes during their break. An improvement to this analysis could have been taking the findings of the conducted studies, and perform a new bespoke experiment, comparing the results of different intervals of rest period between control groups. I also uncovered that the country in which these case studies were selected from was not mentioned, only that they were taken in English. Perhaps a comparative study could be made between individuals form different regions of the world, like Asia and Europe, to see if there were any noticeable differences in performance when micro-breaks were given. It shared a similar weakness from the first study reviewed, where diversity was lacking. But I do agree with the findings that micro-breaks are needed in order for an individual to remain productive, as based off the research above, an individual’s performance is tied to how well he or she manages their time and workload.

This literature review analyzed two studies from 2015 and 2022, emphasizing the importance of time management today. The findings showed that practices like planning and intermittent breaks enhance performance in tasks such as studying or working. This supports the need for a task manager app incorporating the 52/17 technique, which aids students in focusing and planning study schedules with timed work and breaks.

The Effectiveness of Goal Visualization

Cheema and Bagchi[4] explored how visualizing goals impacts an individual's sustained effort towards achieving objectives. Their study showed that intuitive, short-term goals are perceived as easier to reach, especially when broken into sub-tasks. Visualization was found to make goals appear nearer and increase effort and efficiency in pursuing them, highlighting the power of visualization and its effectiveness when goals are clearly defined.

A total of five studies were conducted to assess the effects of goal visualization.

The first study took 68 Olympic swimmers and had them swim 30 laps, each of which was a 100m. The measurements used was to take the difference in swim times of the first 50m, when the swimmer was facing away from the finish line, and the last 50m, when the swimmer was able to see the finish line, hence, able to visualize the end goal. The results revealed that the time difference decreased as the swimmer neared the finish line, showing a positive influence on the swimmer when he was approaching the goal, despite the presence of fatigue.

The second study had 79 students sustain grip pressure for 130 seconds, split into two groups. One group viewed a horizontal progress bar, while the other saw a stopwatch at 30-second intervals. Results showed that the group with the stopwatch exerted significantly less pressure towards the end, whereas those with the progress bar maintained more uniform pressure. This suggests that visualizing goal proximity helps sustain effort.

The third study took 183 undergraduates and tasked them to save $750 for a fabricated vacation to Europe. There were two groups, one that could easily visualize their savings with a horizontal bar that was shaded depending on how much was already saved, 30% or 70%, whilst the other control group was only given textual representation of the savings amounts. The measurements used was a scale given to the participants to assess their commitment to reaching the required amount. The scale was based off Wright and Kacmar[5]. The results revealed those given the bar that made visualizing the amount left to save up, committed greater effort to saving than those that had a difficult time visualizing.

Just off these three studies, participants with goal visualization, like a progress bar, outperformed those without it, in endurance tests. These findings suggest that visualizing goals boosts effort and motivation, supporting the feature a project management tool with visual feedback, such as a progress bar, to track progress and remaining tasks.

However, the studies conducted above do leave room for question as to the factors that could contribute to how likely an individual is able to outperform another, given that both has access to easy visualization tools. For example, the undergraduates whose grip strength was tested, could house bias to the strength of the individual. A student with phenomenal grip strength and endurance could exert the most amount of willpower and willingness to complete the test, regardless of what goal visualization tool he was given. Thus, it is important to note an individual's characteristics and personality might be a contributing factor when to comes to accomplishing goals.

Comparison of Existing Applications

The following is a review of similar applications that have already been published on the Google Play Store. These competitors were selected for analysis due to their popularity for Android Devices.

Microsoft To Do[6] is a to-do list mobile application developed by Microsoft for Android and IOS devices(Figure A1). The analysis of the application are as follows:

1. Users must sign in or create an account to use the application, which may drive them to competitor apps if they don't wish to log in[7].

2. The app lacks a tutorial, making it hard for users to discover all features.

3. The application's aesthetic could be improved. The contrast between the black top bar and white main container, affects readability of the text, and contrast of colors is one of the key areas in GUI design[8].

4. Wang and colleagues[9] noted that students face significant digital distractions. A focus tool to help manage time and minimize these distractions, as discussed in the Time Management section, would be beneficial.

5. A study done by the U.S. Department of Labor[10] showed that 78% of the population, spent their days engaged in household activities while 44% of the population were engaged in work/work related activities. This indicates users have many tasks to accomplish. The application lacks a project management tool to meet this need.

Google Tasks[11] integrates seamlessly with other Google services like Gmail and Google Calendar, allowing users to create, view, and manage tasks across multiple platforms (Figure A2). The analysis of the application are as follows:

1. Google allows users to access basic to-do list features without signing in. Users can log in to sync tasks with Gmail and Google Calendar.

2. Has no tutorial, making it difficult for users to uncover all the available features.

3. Google has adopted is the flat UI approach[12]. This provides a clean and easy interface to navigate.

4. A tool that could aid students in focusing and managing their time, as discussed in the Time Management section, could prove fruitful.

5. No tool for users to undertake and manage large scale projects, and houses no way to visualize the progress being made.

6. Does not allow users to upload and attach images as part of their notes.

Tasks by Pocket Brilliance[13]. A review of this application reduces the bias of only analyzing task manager applications from already established technology giants (Figure A3).
The analysis of the application are as follows:

1. It uses minimal API such as calendar, syncs the application and its data across multiple devices. It does not require users to log in. However, it requires users to purchase the premium version of the application, in order to synchronize across devices.

2. This application does not collect or share user’s data, a good practice to entice users to use their application, as data collection from corporations is rampant in today’s world[14].

3. Has a poor tutorial that only activates once users start to interact with the application. It does not give the option to replay the tutorial in the settings page, which is extremely cluttered.

4. Adopted a flat UI approach.

5. No tool that could aid students in focusing and managing their time, as discussed in the Time Management section.

6. No tool for users to undertake and manage large scale projects, and houses no way to visualize the progress being made.

Key Takeaways

The findings uncovered from the analysis, that will be carried over to my application, are highlighted under the "Carry Over" column. These features were selected to provide the best experience to users, whilst eliminating potential inconveniences. Moreover, the literature review conducted on time management and goal visualization provides the justification for the intended features of the application, such as the need for a project management and focus tool.

Name of Application	Good Practices	Bad Practices	Carry Over
Microsoft To Do	- Allows customization via images	- Forcing users to sign in - Bad design principle - No tutorial - No focus tool - No project tool	- Include focus & project tool - Include a tutorial - Do not force users to sign in - Select proper design principles
Google Tasks	- Great UI design choice, flat principle	- No tutorial - No focus tool - No project tool - Can't attach images	- flat principle and minimalism design - Ability to attach images
Tasks by Pocket Brilliance	- Minimalize use of 3rd party API - No ads or data collection - Minimalist and flat UI approach	- No focus tool - No project tool - Overwhelming settings page	- Use 3rd party services only when necessary - Do not collect data or house advertisements - Adopt a minimalist approach

Project Design

Domain & Users

The domain of the project falls under productivity, and the mobile application being built in this project is intended for students and adults. I intend to develop, test, and deliver an Android application to the Google Play Store. The reason I will not be developing for IOS devices, and attempt to publish it into the Apple App Store, is due to the fact that I do not own any Apple devices, particularly an iPhone. Therefore, I am limited by my hardware.

Justification of Selected Features

The modern world is increasing in terms of responsibilities, and digital distractions. A task manager application that employs time management techniques, has shown to improve student's academic performance, whilst goal visualization, has shown to help adults such as athletes and managers, to exert more effort to reach their end goal, based on the current literature.

Therefore, to address time management, the task manager application will house a timer, employing the 52/17 technique[15]. An individual will time themselves to work for 52 minutes and rest for 17 minutes. And in regards to goal setting and visualization, the application will house a project management tool that will allow users to undertake large sets of tasks, whilst being able to visualize their progress, via a horizontal progress bar, providing motivation to see the project through to end, as discussed in the literature review. The application must also house the ability to use images as notes. Analysis of existing application has also shown tutorials, or the lack thereof, emphasizing the need for a proper tutorial, guiding users to all the features available to them.

Selected Features based off the Literature Review
Project Management Tool - Allow management and prioritizing of sub-tasks, notes, and provides visual feedback
Focus Tool - timer to allow users to manage their time spent working and resting
Checklist - to-do list with calendar integration
Notes - note section that gives users freedom to express their thought, via text or images
Tutorial - Guide users on all the features available to them

Technology & Methodology

The chosen technology to build this mobile application is React Native[16] and Expo[17]. React Native is a JavaScript framework developed by Meta[18], with the aim of allowing developers to use a single language, in this case JavaScript, to develop applications for a particular platform or operating system using the platform's own tools and languages.

Expo is a framework and platform built around React Native that simplifies mobile app development, building, and deployment. It provides a managed workflow with a suite of pre-configured libraries and APIs, enabling me to start building apps quickly without dealing with native code configurations. The Expo CLI and Expo Go app streamlines development, allowing for instant previewing on physical devices. Furthermore, the Expo EAS cloud service will be used to build and deploy the application. React Native based mobile application, can also run on IOS devices, giving this project the room for upgradability to Apple devices, should one come into my possession for testing, in the future. Testing will be done using the Jest Framework[19].

The key APIs and libraries that were used to build this application are as follows:

- @react-native-community/datetimepicker
- @react-navigation/bottom-tabs
- @react-navigation/stack
- expo
- expo keepawake
- expo-camera
- expo-image-picker
- expo-notifications
- expo-sqlite
- react
- react-native
- react-native-calendars
- react-native-countdown-circle-timer
- react-native-progress
- react-native-reanimated
- react-native-tableview-simple
- jest

For a full list of libraries used, refer to the package.json file in the source code.

The software development for this project will utilize Sprints[20] and User-Centered Design (UCD)[21], both methodologies stemming from Agile software development. I selected these approaches for their effectiveness in addressing the project's requirements and final product characteristics. Sprints are repeatable stages within a software development cycle, while UCD focuses on the needs, preferences, and limitations of end-users through active feedback and iterative testing.

Given that the task manager mobile application prioritizes user experience over complex technical features, UCD enables a better understanding of users and allows for iterative improvements based on their feedback. This approach ensures the application consistently meets users' needs and prevents biases from my own assumptions about the application's delivery.

Design Structure

Refer to Figure A4 for the user flow diagram and Figure A5 for the relational SQLite database schema.

React Navigation[22] is a core library for modern mobile applications, enabling the creation of multiple components and pages with controlled navigation. It uses a stack data structure, following the Last In, First Out (LIFO) principle, where the most recent item is removed first. The user navigates through the bottom tab, with each tab containing its own page, and the pages are navigated via a stack (Figure A6).

The project tab tool, encompasses not only the React Navigation library but also APIs from Expo as mentioned in the Technology & Methodology section, that give access to the Android Software Development Kit(SDK).

The project tool feature will include an overview page showing project titles and their progress based on completed tasks. Tapping a project navigates to a details page where users can create and view tasks, subtasks, and notes. The SQLite API will be utilized to perform CRUD operations to manage storing and displaying project information, tasks, subtasks, notes, and images on a local database. Because this is done locally, users will not require an internet connection(Figure A7).

The task/note creation page of the project tool will contain a text input for users to enter the task, as well as multiple additional cells below to provide more useful functionalities such as deadline for the task, subtasks for the main task, and reminder to complete the task by the due date, and camera option to attach images.

To set a deadline, a Calendar library will be used for date selection, which is then stored in the local SQLite database. The Expo Notifications API will send reminders if set by the user. For image capture or import, the Expo Camera and ImagePicker APIs will access the device’s camera and gallery, storing the images in the SQLite database.

The focus timer tool will feature a circular progress bar to display the remaining time, emphasizing goal visualization and visual feedback, as supported by the literature review. Below the timer, start and stop buttons will initiate the countdown. To prevent the device from sleeping and ensure continuous visual feedback, the KeepAwake API will be used, allowing the user to stay motivated by seeing how much time has elapsed and how much remains (Figure A8).

The literature review on goal visualization and time management, along with analysis of existing apps, justifies the project and focus tool. Adults have a myriad of tasks to accomplish on a daily basis, as cited from the US Department of Labor study. As technological integration grows, students seek better tools for academic tasks, despite the consensus that the tools available to them could be improved upon[23]. Thus, the project management tool and its features cater and address the needs of my target audience of students and adults, providing a means to manage tasks, projects, and productivity techniques such as time management.

Low & High-Fidelity Wireframes

Low and high-fidelity wireframes were developed for use in the first round of user-based testing in the Development phase, as well as to gather initial feedback on the design and prototype features(Figure A9 & A10).

Usability is an important aspect in the design of this application. To ensure a smooth experience, especially visually, the UI must possess simplicity and clarity such that the interface is kept as simple as possible, whilst avoiding clutter. For research shows that majority of mobile users prefer a minimalist approach in regards to the interface[24]. Examples of this design choice and UI are the bottom navigation tabs, buttons and page title, so that users know exactly where they are and they can do, in terms of pressable elements.

I gathered user feedback on the wireframes through Google Surveys. Seven students and staff from the Singapore Institute of Management (SIM) participated, providing valuable insights for our target audience of students and adults (aged 19-40).

The survey was conducted on site and in person, and no personal data was collected, for privacy purposes. The overall sentiment was neutral to positive (Figure A11). This small survey test has shown that the project and the intended application that is to be developed, is on the right track, with no negative feedback as of yet, on any crucial features.

Application Assets

The necessary icons and vector images will sourced free of charge from Expo Icons[26] and flaticons[27]. Refer to Figure A12 for the application icon and splash screen.

Gantt Chart

A Gantt chart was created to outline the roadmap for this final project (Figure A13).

The key milestones are as follows:

- Completion of Design Phase – June 7th
- Submission of Preliminary Report – June 11th
- Submission of Draft Report – July 22nd
- Completion of Development Phase – August 14th
- Published to Google Play – August 26th
- Submission of Final Project, in its entirety and end of Delivery Phase – September 9th

Implementation

Environment Setup

During the development phase, a local Expo project was created on my machine. Initially, Expo Go was used to view and develop features. Later, the application was built natively for Android using the Expo CLI, running it on my physical device for a more accurate performance representation and debugging. Finally, the project was moved to Expo EAS to build the production version and prepare it for the Google Play Store, as Expo EAS handles the key signing on behalf of the user. The project directory housed all the relevant folders and sub-folders, to modularize the codes (Figure B1). Android Studio was utilized to emulate different Android devices, to ensure the application's layout was responsive.

First Sprint

The first task was to create the React Navigation bottom tab, which would house all the necessary screens for the application. A custom navigation component, situated inside the navigation folder, had to be coded using React components, in order to house the necessary screens in each stacks, as well as customizing the styling and settings icon that would be displayed on the top left of the screen(Figure B2). This was wrapped in a React Context[28] to allow states to passed down the application hierarchy, allowing the styling of the pages to change based on the selected theme, as well as toggling any animation made using React Native Reanimated(Figure B3). In order to set the header styling and color to match the theme selected by the user, custom hooks were created to change the header background and text color, housed appropriately in the hooks folder(Figure B4). Next, the front-end was coded for all the respective screens within each tab. Once the responsive UI was tested via emulators, the focus shifted to implementing the back end of the application.

The SQLite database was initialized and the tables seen in the schema was made with SQL syntax `CREATE TABLE` and exported as a component, enabling the relevant screens to access the tables and perform necessary CRUD operations (Figure B5). To implement the project management tool's visual feedback via a horizontal progress bar, as discussed in the Literature Review and Justification of Selected Features, the following logic was coded. Several React states were created to track the total number of tasks created by the user, the number of tasks left, and a counter for completed tasks. When a user marked a task as completed, that task would be removed from the database, and the progress bar would increment by subtracting 1 (representing 100% completion) from the current number of tasks left, divided by the total number of tasks created by the user. This ensures that the progress value dynamically adjusts as users add or complete tasks throughout the project's lifecycle. The progress value is also stored in the 'progress' column in the 'Projects' table of the relational database. This allows the user to see the project's progress from the home screen and have it update dynamically.

Furthermore, in the task creation screen of the project tool, to enable the user to add a reminder to complete a certain task in a project, push notifications had to be enabled. To register for push notifications on Android, via the Expo workflow, the project had to be first initialized via EAS, which was done in the Environment Setup. At the time of writing, Expo required Firebase Cloud Messaging (FCM) V1 in order to enable push notifications on Android. Thus, a project was created on Firebase[29], a backend cloud computing services by Google. Through the Firebase project, the credential key needed to register the application with Expo, as well as the google services file was generated and configured into the android native module, as well as the Expo Project account credentials settings for the mobile application.

The current state of the application at the end of the first sprint can be seen in Figure B6. At the end of the sprint, user testing was conducted on the application, and feedback (Appendix D) was gathered. Based on this feedback, iterative development was employed to draft improved wireframes(Figure B7), which will be implemented in the second sprint. This iterative process, guided by user-centric design principles, ensures that the application evolves in response to user needs and preferences in the second sprint development.

Second Sprint

Second sprint still in progress…

Throughout the development during the two sprints, a changelog was utilized to document the changes made to the application. Refer to Appendix C for the full list of changes made to the application.

Google Play Store

An application has to be created on the Google Play Console, after creating a developer account, which I have already done. I have decided to name the application TaskTracker 9000, under the developer alias, GammaDigital (Figure C1). There are four stages in the uploading process that must be completed in order to publish the application to the Google Play Store. The internal testing, setting up the app, closed testing, and production.

Internal testing requires me to test the application in beta, which I will do close to the end of the second sprint. Once that is completed, several questionnaires will need to be filled, such as the target audience, does the application contains ads, setting up the description of the application in the Play Store and relevant application screenshots, etc. An important step is setting up the privacy policy of the application, which can be viewed here.

The link to view and download the application in the Google Play Store will be placed here, once it has been uploaded successfully and the development phase completed.

Evaluation

Unit Testing

As mentioned in Technology & Methodology, Jest was used to perform unit testing and it was possible because the code was modularized. Unit testing here was considered white box testing, as the internal codes were known and understood by the tester, me. The tests were preconfigured to run on Android only. Test cases were written for logical functions and all custom components made such as the time converter for the “Focus Tool” and the SQLite database and queries, to name a few. Mocking had to done to simulate certain tests that require dependencies. All test suites made and ran so far have passed.

System Testing

To evaluate the application developed in the first sprint, a system wide component testing was done. Also known as black box testing, as here, I was only interested in the input and output of the components, and whether they were behaving as expected. Each of the 4 tabs of the application, and its relevant features were tested, and the test cases was written and documented in an Excel workbook, that is attached to the source code and on GitHub for viewing. The tester that tested the application in its beta state, bundled and installed into a local Android device, was a student from SIM above the age of 18. The anonymous individual was used as a tester as he had no prior knowledge of the internal workings of the application, and was part of the domain of users the application was intended for, providing an unbiased ground for testing. All test cases passed at time of testing, which was the end of the first sprint.

User Feedback

At the end of each sprint, user feedback was gathered to evaluate the application's performance and usability via students and staff from SIM. This feedback provided valuable insights into areas that required improvement and helped identify aspects of the application that needed adjustment. Refer to Appendix D for the full details of the feedback collected. The feedback highlighted specific points in the flow of the application that the users was pleased or displeased with, such as issues with the user interface, functionality glitches, and any confusion regarding features.

The critical areas were:

1. Users pointed that they wanted to still see the completed project tasks after it was added to the progress bar
2. Users want to have the option to remove the wrongly inputted values in the task creation screen, without having to reload the page.
3. Users wanted to have the freedom to choose the timer time preset.
4. Users wanted to be able to minimize the calendar, to be able to view the to-dos better.

As of now, a fully functioning task manager mobile application has been built and bundled for Android devices via APK installation. It tracks projects, visualizes progress, and includes a focus timer tool. Users can log tasks and notes with image attachments, subtasks, and set reminder notifications and deadlines. The app also integrates a calendar API for marking tasks as completed. The UI is easy to use, with a repayable tutorial to guide them through the features. All information is stored locally on the device, without the need for internet connection, making offline usage possible.

To evaluate the success of the project and the application, and to determine if it has achieved the objective of outperforming current market offerings, an experiment will be conducted with the target audience.

A research conducted by healthcare professionals revealed key methodologies for designing and conducting questionnaires[30], and found that face to face interview provides higher merits like clarification of questions, thus the interview will be conducted on the campus grounds of the Singapore Institute of Management, comprising of students and working adults above the age of 18. Once consent has been granted, participants will be split into three groups. One will use Microsoft's application as seen in the literature review, the other Google, and my own application. They will be given a fabricated project of renovating their home, and have to lists down all the given tasks and notes necessary to accomplish the renovation.

Once they have inputted all the tasks and completed them, I will have them fill out the User Experience Questionnaire[31], to see which application they enjoyed using more. To remove any bias, they respondents will not know that the application was developed by me, until the questionnaire has been completed. No personal data such as names, age or email addresses will be collected, ensuring no ethical breach. The results of the questionnaire will address the objectives set out, and reveal if the task manager application succeeds in the market, leading to a successful final project.

Conclusion

Awaiting proper conclusion after development phase ends.

This project has required me to apply the full breadth of skills and knowledge I have acquired throughout my Computer Science degree. I utilized my testing and debugging skills to ensure the application functions correctly and efficiently. Wireframing and iterative development were essential in designing user-friendly interfaces and continually refining the application based on user feedback. Understanding and implementing the software development lifecycle allowed me to manage the project's progress effectively from conception to deployment. Additionally, I gained hands-on experience in development and production builds, ensuring the application is robust and ready for real-world use. This comprehensive application of my education has solidified my expertise and prepared me for future challenges in software development.

Further work that could be explored is gamifying the focus tool. If the user is able to successfully stay productive for the set duration, he or she is rewarded with a token. A collection or gallery of tokens could be created and rendered. However, should the person exit the application of reset the timer, the token would be lost or destroyed.

References

[1] Nasrullah_PhD, S. and Khan_PhD, M.S. (2015). The impact of time management on the students’ academic achievements. Journal of Literature, Languages and Linguistics, 11, pp.66-71.

[2] Razali, S.N.A.M., Rusiman, M.S., Gan, W.S. and Arbin, N. (2018). The Impact of Time Management on Students’ Academic Achievement. Journal of Physics: Conference Series, [online] 995(1), p.012042. doi:https://doi.org/10.1088/1742-6596/995/1/012042.

[3] Albulescu, P., Macsinga, I., Rusu, A., Sulea, C., Bodnaru, A. and Tulbure, B.T. (2022). ‘Give me a break!’ A systematic review and meta-analysis on the efficacy of micro-breaks for increasing well-being and performance. *PLOS ONE*, [online] 17(8), p.e0272460. doi:https://doi.org/10.1371/journal.pone.0272460.

[4] Cheema, A. and Bagchi, R. (2011). The Effect of Goal Visualization on Goal Pursuit: Implications for Consumers and Managers. Journal of Marketing, 75(2), pp.109–123. doi:https://doi.org/10.1509/jm.75.2.109.

[5] Wright, P. and Kacmar, M. (1994). Goal Specificity as a Determinant of Goal Commitment and Goal Change . [online] www.sciencedirect.com. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0749597884710594#preview-section-cited-by.

[6] Microsoft.com. (2024). To Do List and Task Management App | Microsoft To Do. [online] Available at: https://www.microsoft.com/en-sg/microsoft-365/microsoft-to-do-list-app

[7] Underwood, J. (2017). Learning 001: Sometimes you shouldn’t force user login, even if everyone says you should. [online] Medium. Available at: https://medium.com/@JUnderwood/learning-001-sometimes-you-shouldnt-force-user-login-even-if-everyone-says-you-should-96dd4c416487

[8] Figma (2023). 5 key UI design principles— and how to use them | figma. [online] Figma. Available at: https://www.figma.com/resource-library/ui-design-principles/.

[9] Wang, C.-H., Salisbury-Glennon, J.D., Dai, Y., Lee, S. and Dong, J. (2022). Empowering College Students to Decrease Digital Distraction through the Use of Self-Regulated Learning Strategies. Contemporary Educational Technology, [online] 14(4). Available at: https://eric.ed.gov/?id=EJ1364826.

[10] Bureau Of Labor Statistics (2023). AMERICAN TIME USE SURVEY — 2022 RESULTS. [online] Available at: https://www.bls.gov/news.release/pdf/atus.pdf.

[11] play.google.com. (n.d.). Google Tasks – Apps on Google Play. [online] Available at: https://play.google.com/store/apps/details?id=com.google.android.apps.tasks&hl=en_SG&gl=US&pli=1

[12] DesignerUp. (2023). 6.5 of The Most Popular UI Design Trends and Styles Explained. [online] Available at: https://designerup.co/blog/here-are-6-5-of-the-most-popular-ui-design-trends-and-how-to-design-them/.

[13] play.google.com. (n.d.). Tasks: to do list & tasks – Apps on Google Play. [online] Available at: https://play.google.com/store/apps/details?id=com.tasks.android&hl=en_SG&gl=US

[14] www.privacyjournal.net. (2022). Big-Tech Data Collection: Protect Your Info 2024. [online] Available at: https://www.privacyjournal.net/big-tech-data-collection/.

[15] www.larksuite.com. (n.d.). Unveiling the 52-17 Rule: A Guide to Boosting Productivity. [online] Available at: https://www.larksuite.com/en_us/topics/productivity-glossary/52-17-rule

[16] React Native (2022). React Native · A framework for building native apps using React. [online] reactnative.dev. Available at: https://reactnative.dev/.

[17] Expo. (n.d.). Expo. [online] Available at: https://expo.dev/.

[18] Meta (2024). Meta | Social Metaverse Company. [online] Meta | Social Metaverse Company. Available at: https://about.meta.com/.

[19] Jest (2017). Jest · 🃏 Delightful JavaScript Testing. [online] Jestjs.io. Available at: https://jestjs.io/.

[20] Simplilearn.com. (2020). *Agile Sprint in Software Development: Definition, Process, & Roles*. [online] Available at: https://www.simplilearn.com/agile-sprint-article

[21] Novoseltseva, E. (2019). *User-Centered Design: An Introduction - Usability Geek*. [online] Usability Geek. Available at: https://usabilitygeek.com/user-centered-design-introduction/

[22] reactnavigation.org. (n.d.). React Navigation | React Navigation. [online] Available at: https://reactnavigation.org/.

[23] Dahlstrom, Eden & Bichsel, Jacqueline. (2014). ECAR Study of Undergraduate Students and Information Technology, 2014. 10.13140/RG.2.1.3030.7040.

[24] Sandesara, M., Bodkhe, U., Tanwar, S., Alshehri, M.D., Sharma, R., Neagu, B.-C., Grigoras, G. and Raboaca, M.S. (2022). Design and Experience of Mobile Applications: A Pilot Survey. Mathematics, 10(14), p.2380. doi:https://doi.org/10.3390/math10142380.

[25] react.dev. (n.d.). Rules of Hooks – React. [online] Available at: https://react.dev/reference/rules/rules-of-hooks

[26] icons.expo.fyi. (n.d.). *@expo/vector-icons directory*. [online] Available at: https://icons.expo.fyi/Index.

[27] Flaticon (2024). *Flaticon, The Largest Database of Free Vector Icons*. [online] Flaticon. Available at: https://www.flaticon.com/.

[28] react.dev. (n.d.). *createContext – React*. [online] Available at: https://react.dev/reference/react/createContext.

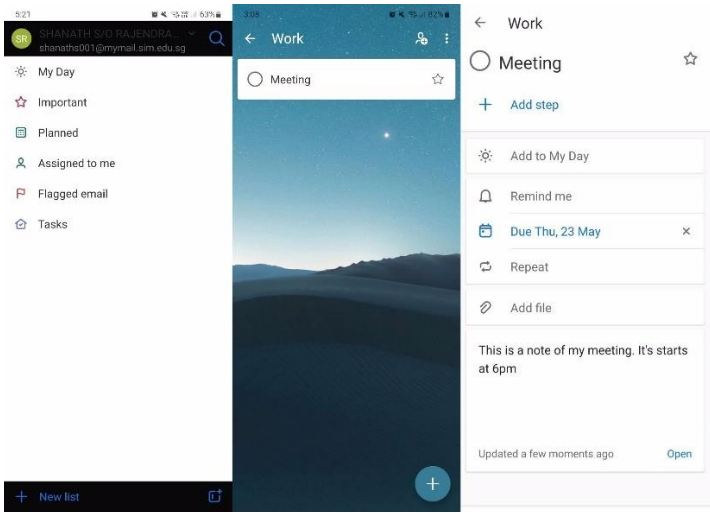
[29] Google (2023). *Firebase*. [online] Firebase. Available at: https://firebase.google.com/.

[30] Roopa, S. and Rani, M. (2012). Questionnaire Designing for a Survey. The Journal of Indian Orthodontic Society, [online] 46(4), pp.273–277. Available at: https://journals.sagepub.com/doi/pdf/10.5005/jp-journals-10021-1104.

[31] UEQ (2018). User experience questionnaire (UEQ). [online] Ueq-online.org. Available at: https://www.ueq-online.org/.

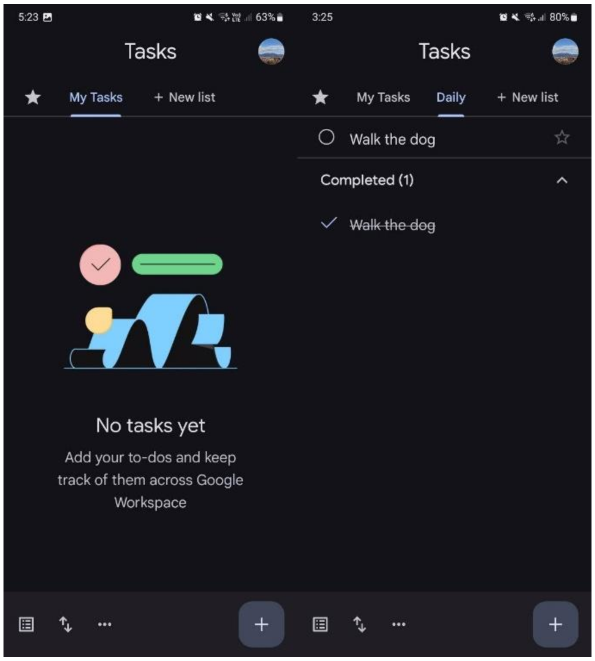
Appendices

Appendix A



This is a note of my meeting. It's starts at 6pm

Figure A1Microsoft To Do



No tasks yet

Add your to-dos and keep track of them across Google Workspace

Figure A2 Google Tasks

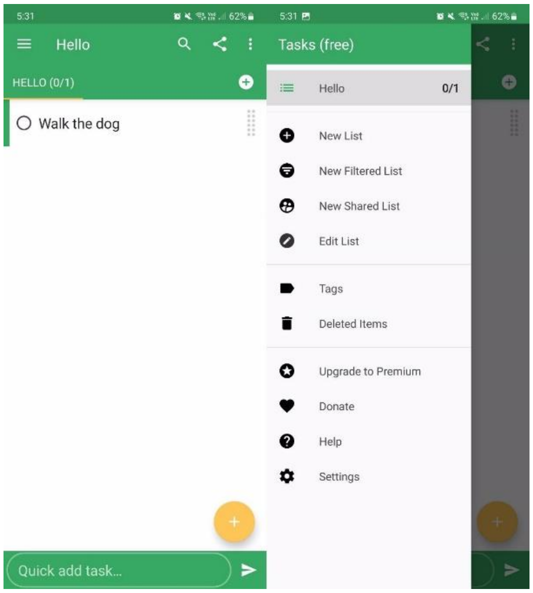


Figure A3 Tasks by Pocket Brilliance

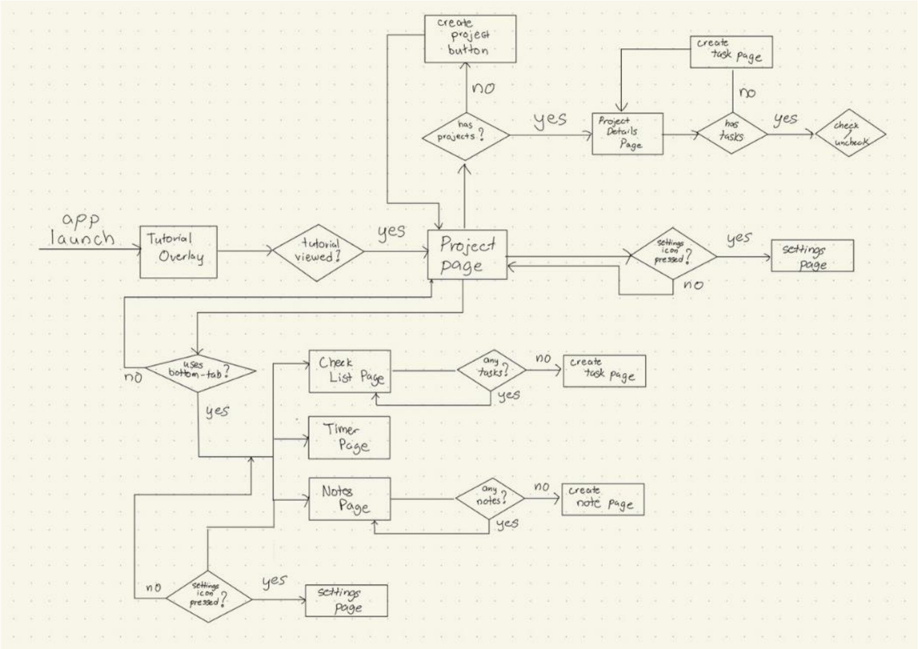
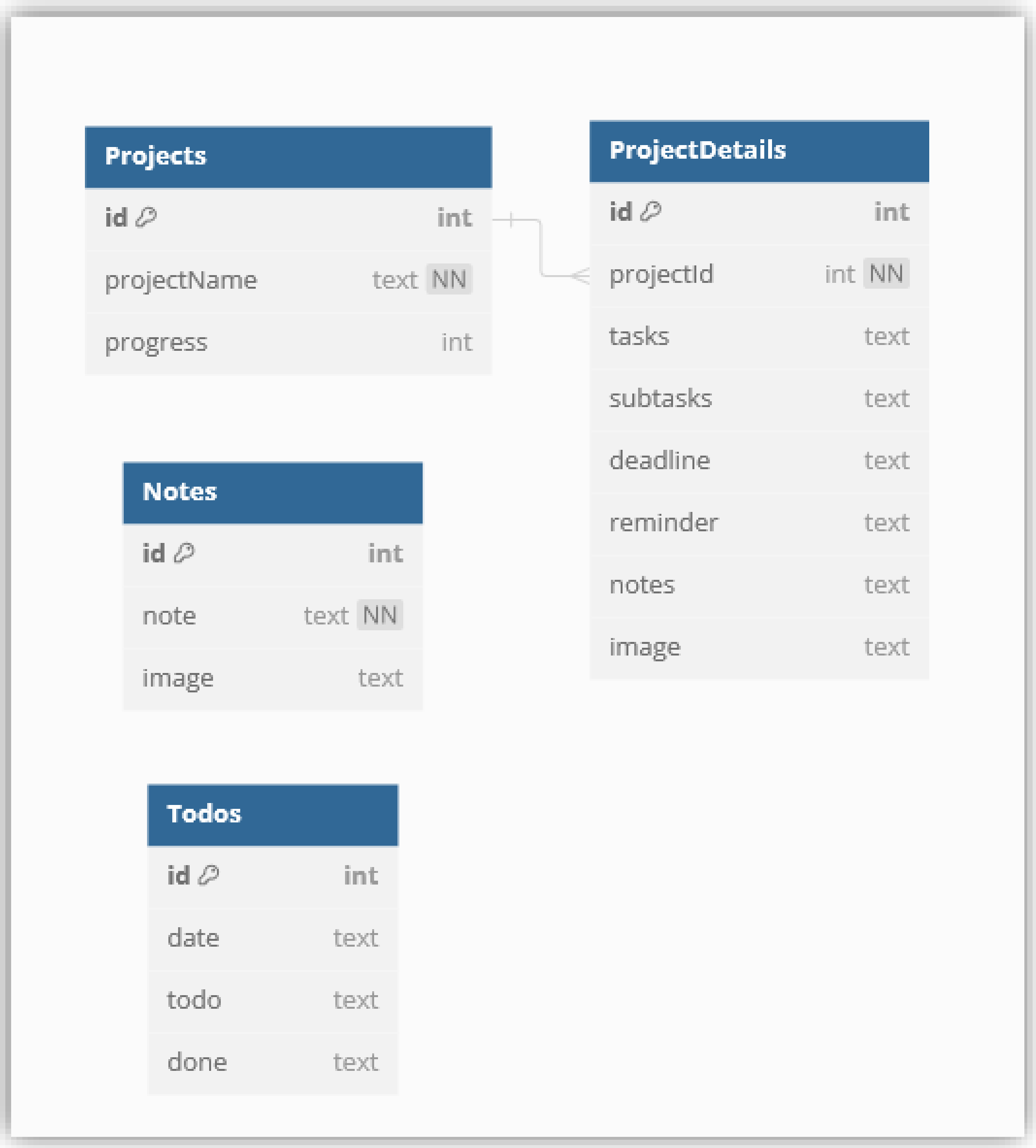


Figure A4 User Flow



Projects	
id	int
projectName	text NN
progress	int

ProjectDetails	
id	int
projectId	int NN
tasks	text
subtasks	text
deadline	text
reminder	text
notes	text
image	text

Notes	
id	int
note	text NN
image	text

Todos	
id	int
date	text
todo	text
done	text

Figure A5 Relational Database Schema

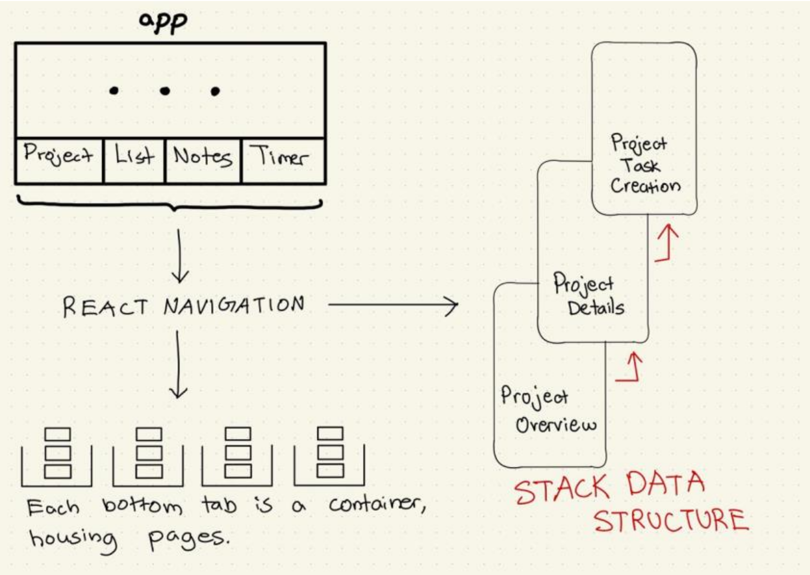


Figure A6 React Navigation Overview

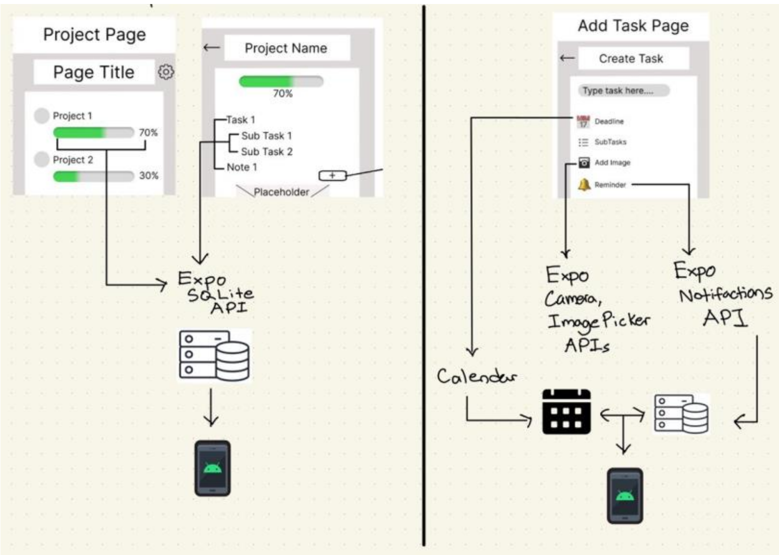


Figure A7 Project Tab Structure

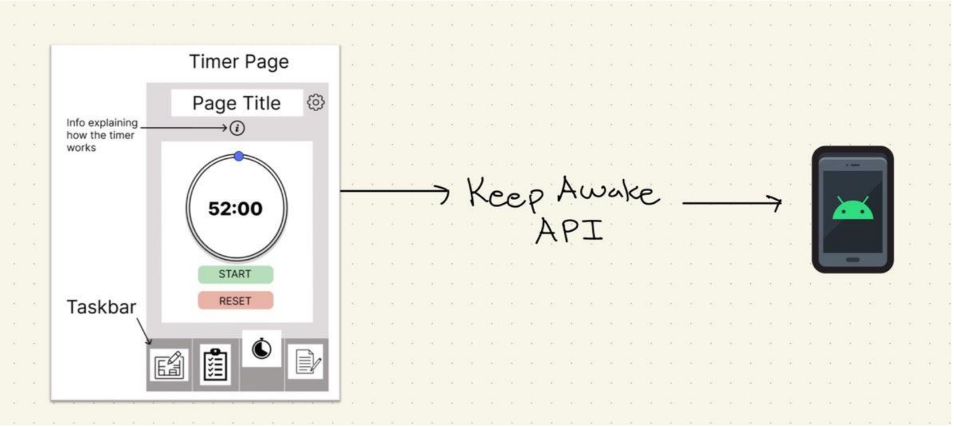


Figure A8 Focus Tool Overview

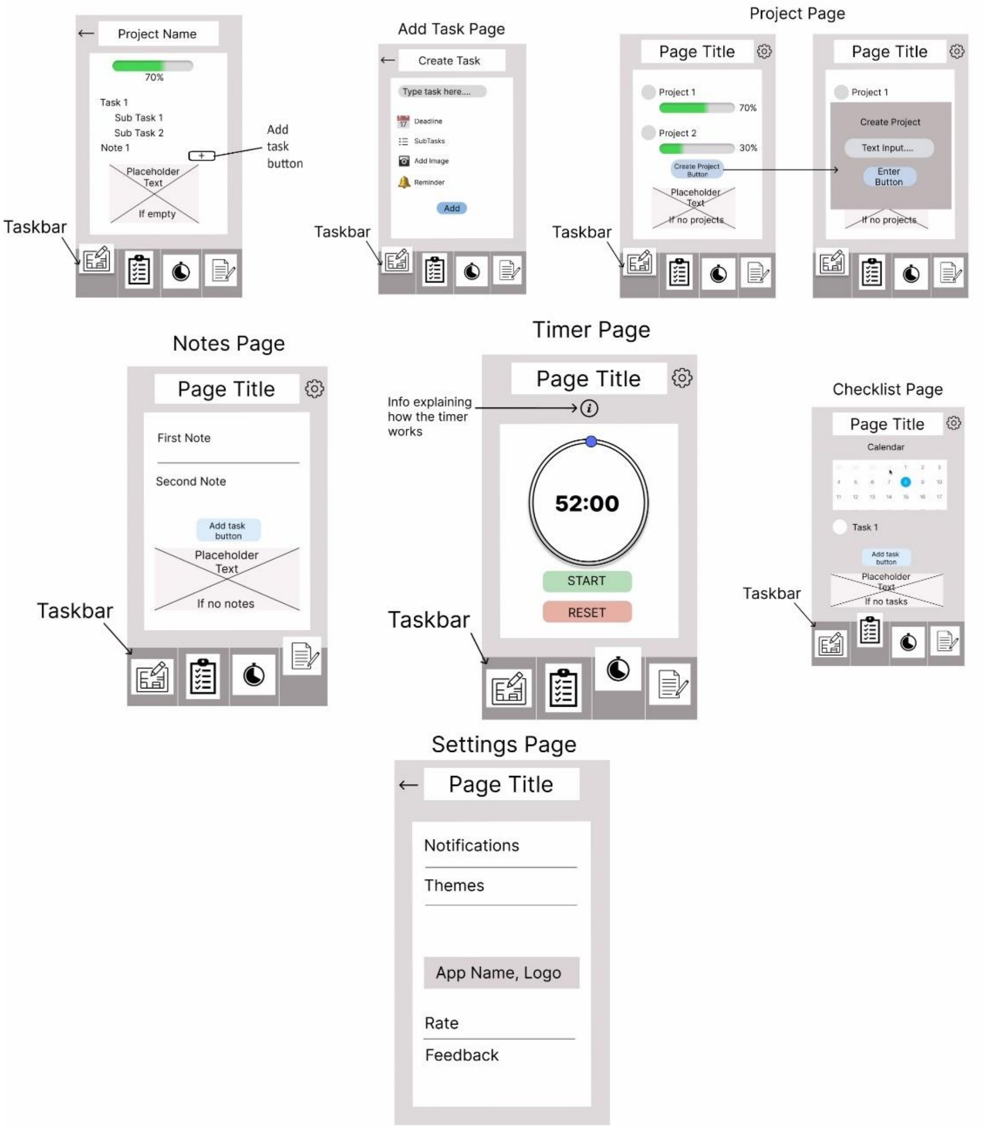


Figure A9 Low Fidelity Wireframe

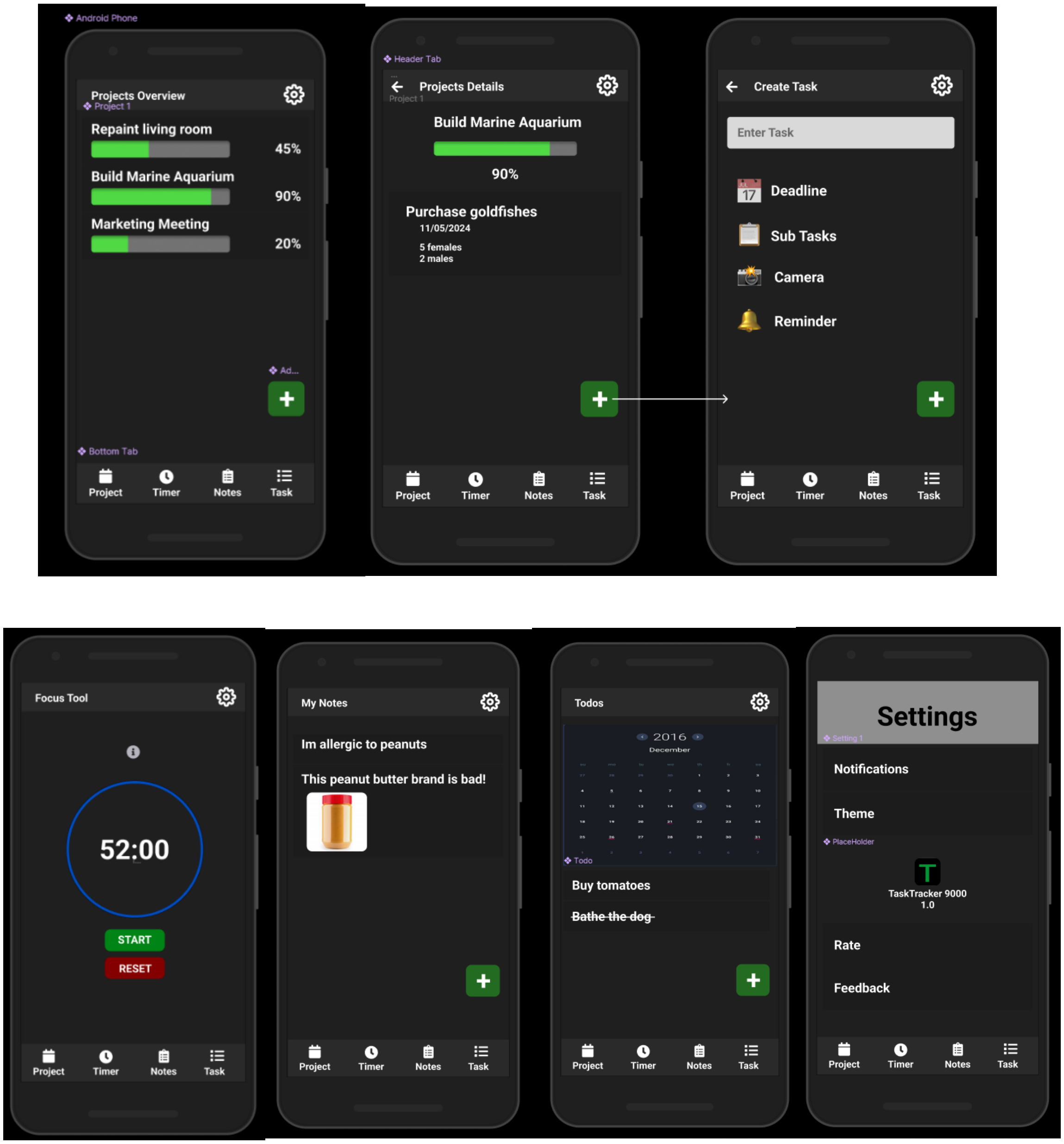
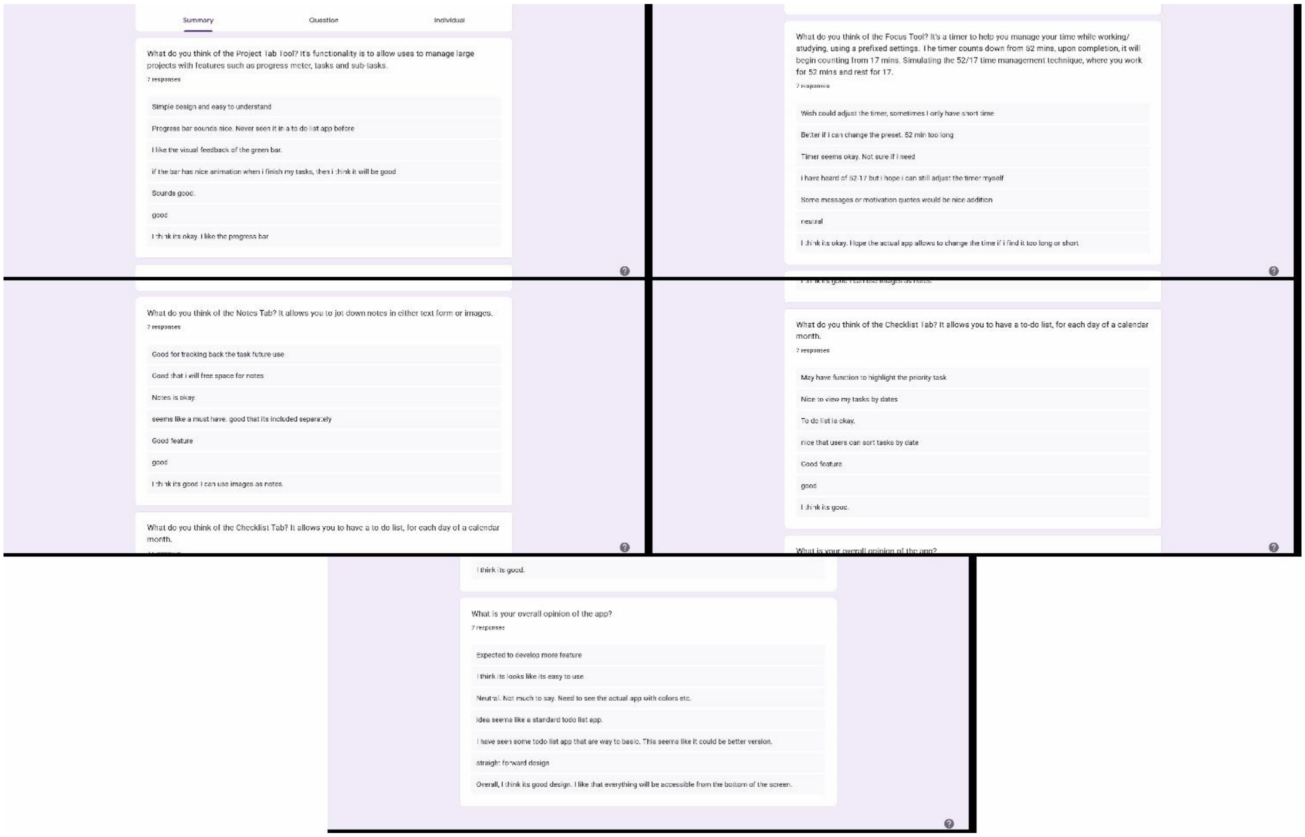


Figure A10 High Fidelity Wireframe



What do you think of the Project Tab Tool? It's functionality is to allow uses to manage large projects with features such as progress meter, tasks and sub-tasks.

7 responses

Simple design and easy to understand

Progress bar sounds nice. Never seen it in a to do list app before

I like the visual feedback of the green bar.

if the bar has nice animation when i finish my tasks, then i think it will be good

Sounds good.

good

I think its okay. I like the progress bar

What do you think of the Focus Tool? It's a timer to help you manage your time while working/ studying, using a prefixed settings. The timer counts down from 52 mins, upon completion, it will begin counting from 17 mins. Simulating the 52/17 time management technique, where you work for 52 mins and rest for 17.

7 responses

Wish could adjust the timer, sometimes I only have short time

Better if i can change the preset. 52 min too long

Timer seems okay. Not sure if i need

i have heard of 52-17 but i hope i can still adjust the timer myself

Some messages or motivation quotes would be nice addition

neutral

I think its okay. Hope the actual app allows to change the time if i find it too long or short

What do you think of the Notes Tab? It allows you to jot down notes in either text form or images.

7 responses

Good for tracking back the task future use

Good that i will free space for notes

Notes is okay.

seems like a must have. good that its included separately

Good feature

good

I think its good. I can use images as notes.

What do you think of the Checklist Tab? It allows you to have a to do list, for each day of a calendar month.

What do you think of the Checklist Tab? It allows you to have a to-do list, for each day of a calendar month.

7 responses

May have function to highlight the priority task

Nice to view my tasks by dates

To do list is okay.

nice that users can sort tasks by date

Good feature

good

I think its good.

What is your overall opinion of the app?

I think its good.

What is your overall opinion of the app?

7 responses

Expected to develop more feature

I think its looks like its easy to use

Neutral. Not much to say. Need to see the actual app with colors etc.

idea seems like a standard todo list app.

I have seen some todo list app that are way to basic. This seems like it could be better version.

straight forward design

Overall, I think its good design. I like that everything will be accessible from the bottom of the screen.

Figure A11 Google Survey

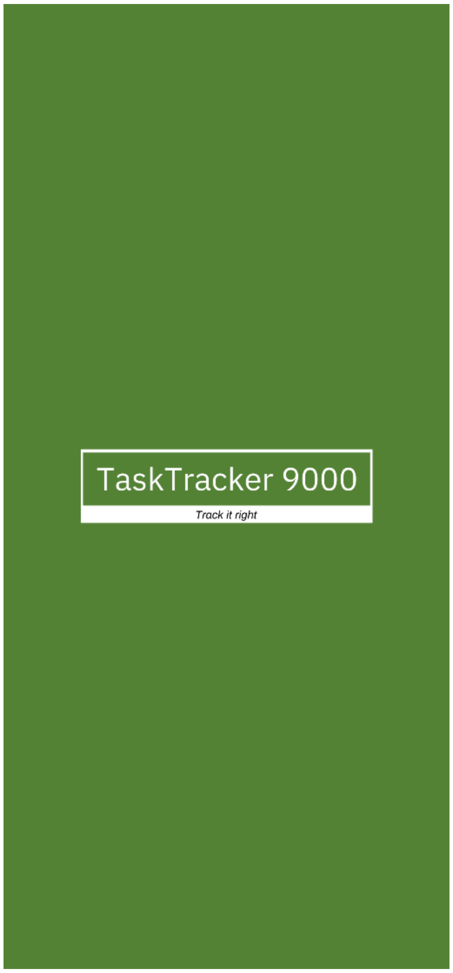


Figure A12 Splash Screen

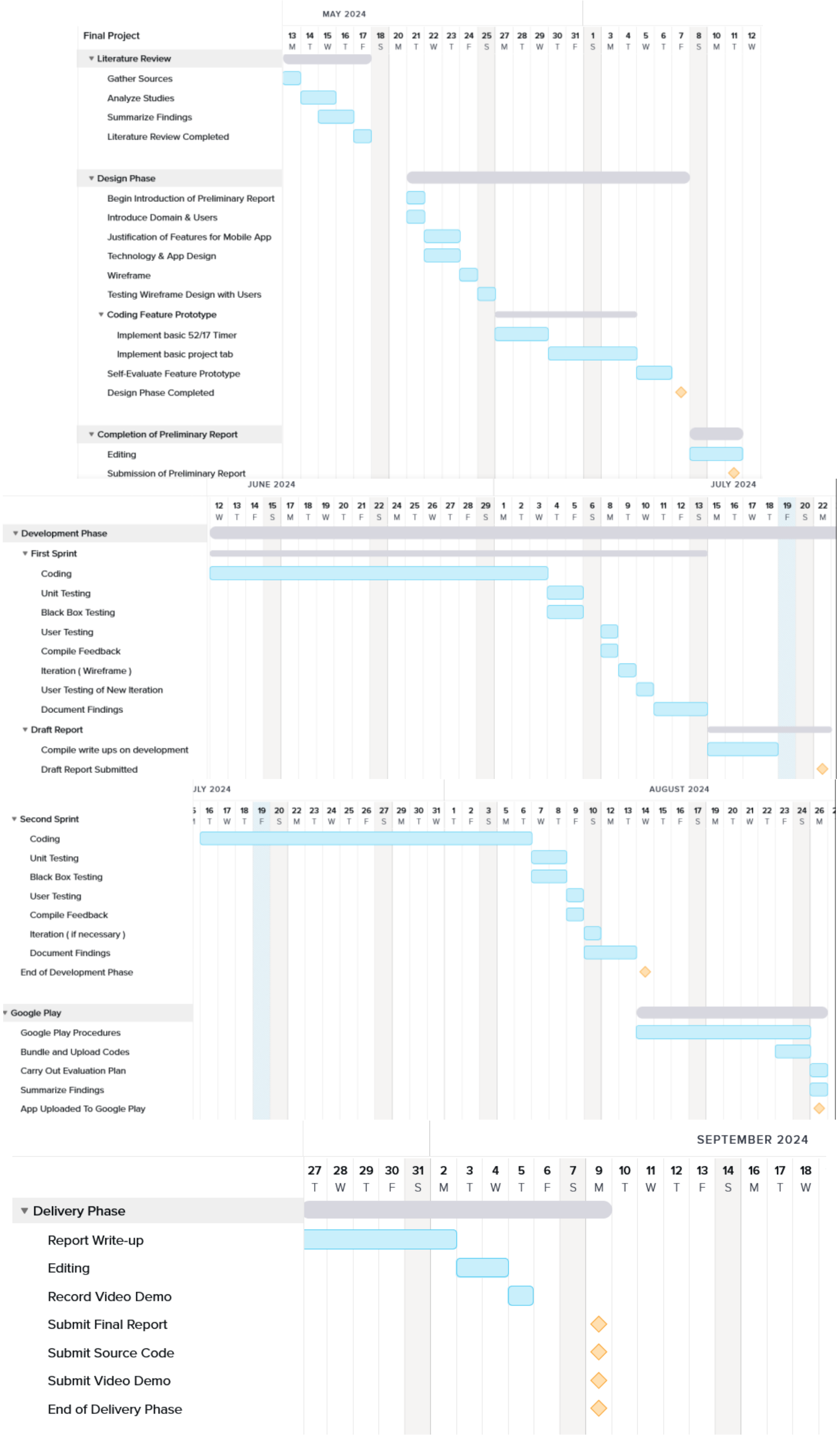
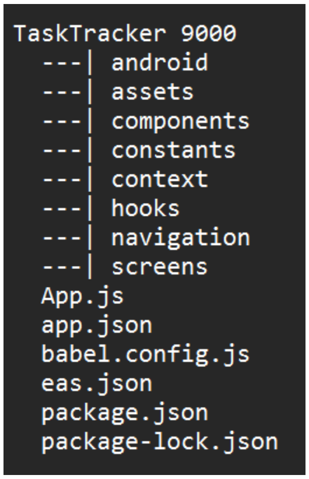


Figure A13 Gantt Chart

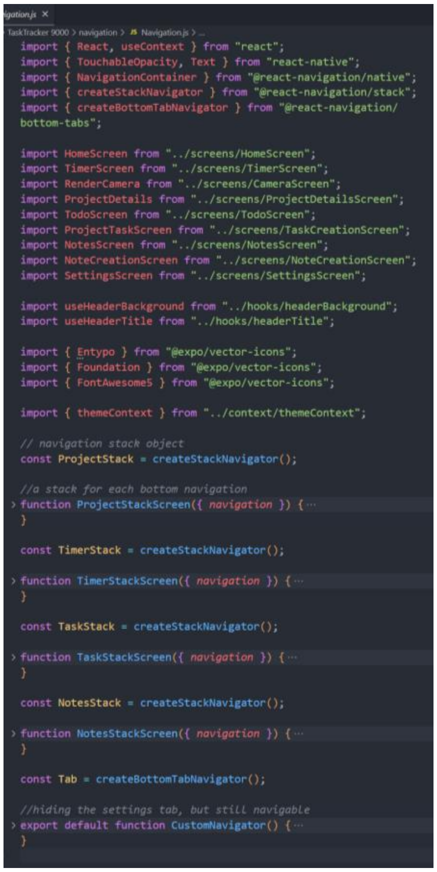
Appendix B

The contents of this Appendix include the code snippets and iteration of the wireframes that were mentioned in Sprint sections.



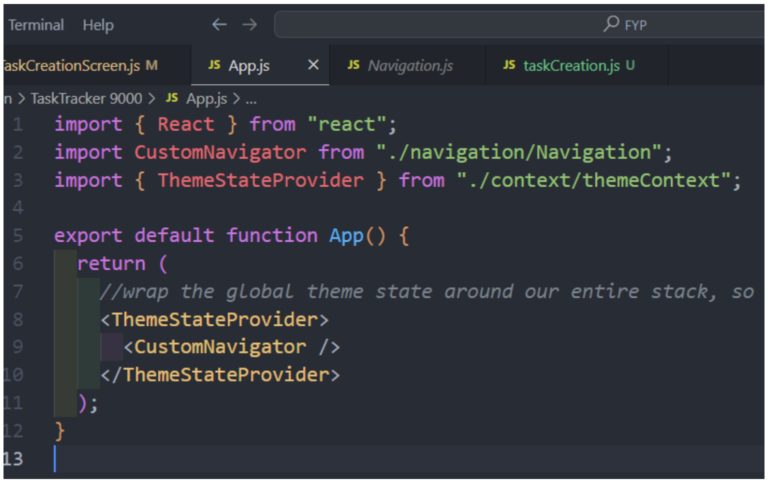
```
TaskTracker 9000
  ---| android
  ---| assets
  ---| components
  ---| constants
  ---| context
  ---| hooks
  ---| navigation
  ---| screens
  App.js
  app.json
  babel.config.js
  eas.json
  package.json
  package-lock.json
```

Figure B1 File Directory



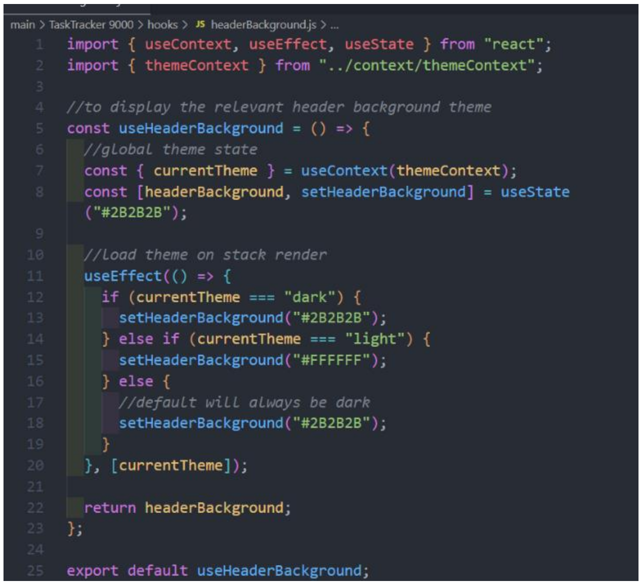
```
import { React, useContext } from "react";
import { TouchableOpacity, Text } from "react-native";
import { NavigationContainer } from "@react-navigation/native";
import { createStackNavigator } from "@react-navigation/stack";
import { createBottomTabNavigator } from "@react-navigation/
bottom-tabs";

import HomeScreen from "../screens/HomeScreen";
import TimerScreen from "../screens/TimerScreen";
import RenderCamera from "../screens/CameraScreen";
import ProjectDetails from "../screens/ProjectDetailsScreen";
import TodoScreen from "../screens/TodoScreen";
import ProjectTaskScreen from "../screens/TaskCreationScreen";
import NotesScreen from "../screens/NotesScreen";
import NoteCreationScreen from "../screens/NoteCreationScreen";
import SettingsScreen from "../screens/SettingsScreen";

import useHeaderBackground from "../hooks/headerBackground";
import useHeaderTitle from "../hooks/headerTitle";

import { Entypo } from "@expo/vector-icons";
import { Foundation } from "@expo/vector-icons";
import { FontAwesome5 } from "@expo/vector-icons";

import { themeContext } from "../context/themeContext";

// navigation stack object
const ProjectStack = createStackNavigator();

//a stack for each bottom navigation
function ProjectStackScreen({ navigation }) {⋯
}

const TimerStack = createStackNavigator();

function TimerStackScreen({ navigation }) {⋯
}

const TaskStack = createStackNavigator();

function TaskStackScreen({ navigation }) {⋯
}

const NotesStack = createStackNavigator();

function NotesStackScreen({ navigation }) {⋯
}

const Tab = createBottomTabNavigator();

//hiding the settings tab, but still navigable
export default function CustomNavigator() {⋯
}
```

Figure B2 Custom Navigator Component



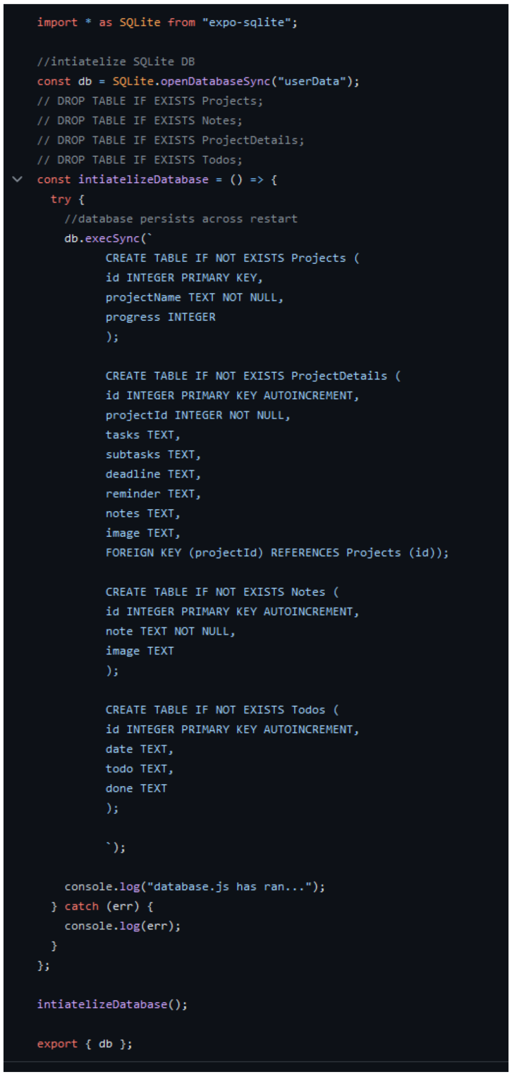
```
import { React } from "react";
import CustomNavigator from "./navigation/Navigation";
import { ThemeStateProvider } from "./context/themeContext";

export default function App() {
  return (
    //wrap the global theme state around our entire stack, so
    <ThemeStateProvider>
      <CustomNavigator />
    </ThemeStateProvider>
  );
}
```

Figure 13 Global Context



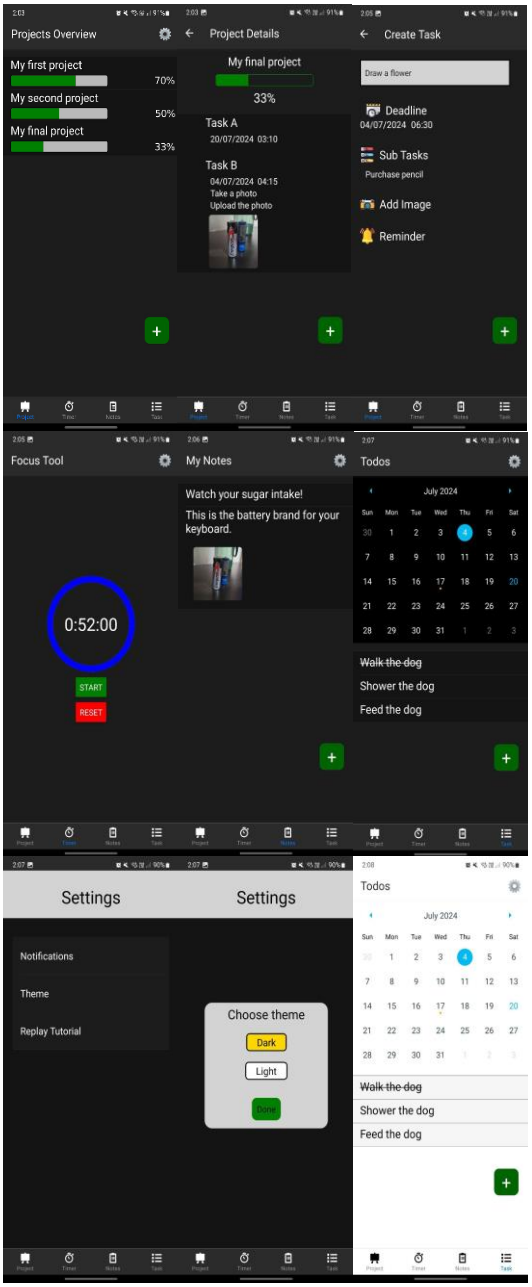
```
import { useContext, useEffect, useState } from "react";
import { themeContext } from "../context/themeContext";

//to display the relevant header background theme
const useHeaderBackground = () => {
  //global theme state
  const { currentTheme } = useContext(themeContext);
  const [headerBackground, setHeaderBackground] = useState("#2B2B2B");

  //load theme on stack render
  useEffect(() => {
    if (currentTheme === "dark") {
      setHeaderBackground("#2B2B2B");
    } else if (currentTheme === "light") {
      setHeaderBackground("#FFFFFF");
    } else {
      //default will always be dark
      setHeaderBackground("#2B2B2B");
    }
  }, [currentTheme]);

  return headerBackground;
};

export default useHeaderBackground;
```

Figure B4 Header Custom Hooks



```
import * as SQLite from "expo-sqlite";

//intiatelize SQLite DB
const db = SQLite.openDatabaseSync("userData");
// DROP TABLE IF EXISTS Projects;
// DROP TABLE IF EXISTS Notes;
// DROP TABLE IF EXISTS ProjectDetails;
// DROP TABLE IF EXISTS Todos;
const intiatelizeDatabase = () => {
  try {
    //database persists across restart
    db.execSync(`
          CREATE TABLE IF NOT EXISTS Projects (
          id INTEGER PRIMARY KEY,
          projectName TEXT NOT NULL,
          progress INTEGER
          );

          CREATE TABLE IF NOT EXISTS ProjectDetails (
          id INTEGER PRIMARY KEY AUTOINCREMENT,
          projectId INTEGER NOT NULL,
          tasks TEXT,
          subtasks TEXT,
          deadline TEXT,
          reminder TEXT,
          notes TEXT,
          image TEXT,
          FOREIGN KEY (projectId) REFERENCES Projects (id));

          CREATE TABLE IF NOT EXISTS Notes (
          id INTEGER PRIMARY KEY AUTOINCREMENT,
          note TEXT NOT NULL,
          image TEXT
          );

          CREATE TABLE IF NOT EXISTS Todos (
          id INTEGER PRIMARY KEY AUTOINCREMENT,
          date TEXT,
          todo TEXT,
          done TEXT
          );

          `);

    console.log("database.js has ran...");
  } catch (err) {
    console.log(err);
  }
};

intiatelizeDatabase();

export { db };
```

Figure B5 SQLite database file



My first project 70%

My second project 50%

My final project 33%

My final project

33%

Task A
20/07/2024 03:10

Task B
04/07/2024 04:15
Take a photo
Upload the photo

Draw a flower

Deadline
04/07/2024 06:30

Sub Tasks
Purchase pencil

Add Image

Reminder

0:52:00

START

RESET

Watch your sugar intake!

This is the battery brand for your keyboard.

July 2024

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

~~Walk the dog~~

Shower the dog

Feed the dog

Notifications

Theme

Replay Tutorial

Choose theme

Dark

Light

Done

Figure B6 Sprint Live App

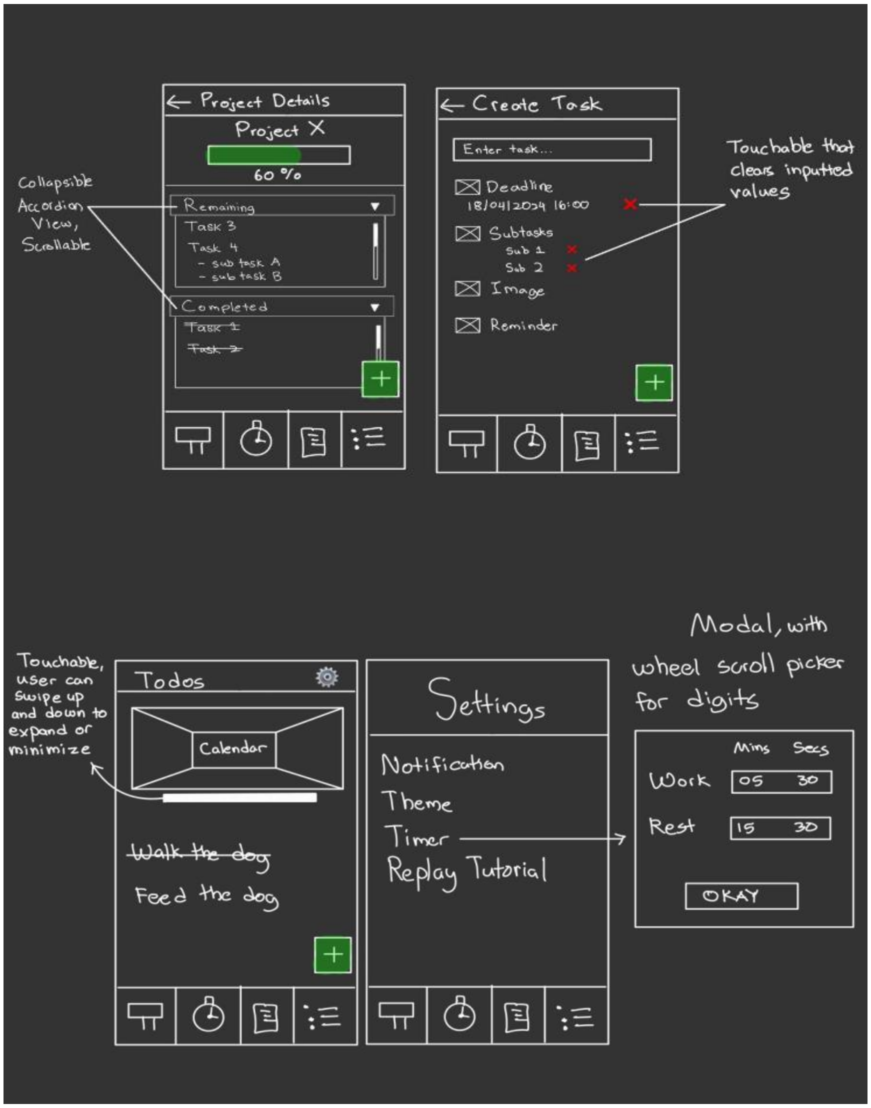
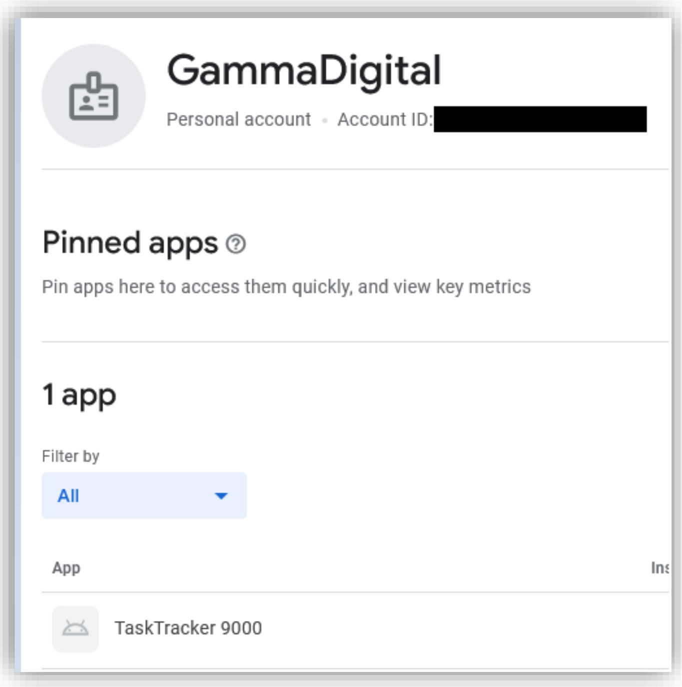


Figure B7 Low Fidelity Iterated Wireframe

Improved project details, task creation, todos and settings screens.

Appendix C

Process of uploading the application to the Google Play Store.



GammaDigital

Personal account • Account ID: [redacted]

Pinned apps

Pin apps here to access them quickly, and view key metrics

1 app

Filter by

All

App	Ins
TaskTracker 9000	

Figure C1 Google Play Console

Appendix D

This appendix houses the changelog and user feedback gathered throughout the project

User Feedback

Below is a compilation of the feedback gathered from the face-to-face user testing.

- The application icon needed rework
- Long pressing the cells to delete the cells were preferred, users did not like a delete icon to delete the project or cell
- Many wanted the ability to view already finished project tasks, instead of deleting them away instantly
- Collapsible accordion needed to be added
- Some liked the 52/17 preset
- Most preferred to be able to manually set the timer
- Most appreciated the ability to add images as attachments.
- Most highlighted calendar size was appropriate, but the todo view was rather small
- Had to increase todo view, by making calendar minimizable.
- A delete cross was needed at the task creation screen for the project tool, users did not appreciate reloading the page to re-correct their false input.
- Todo tasks that were completed should have a greyed-out color, in addition to the strikethrough.
- Audio or vibration was needed when the timer was up

# Changelog

All notable changes to this project will be documented in this file.

## [First Sprint]

### Added

- redid app logo designed, did not fit properly
- added marked dots to calendar, to signify which dates have todos
- added long press on cells to delete

### Fixed

- fixed application crashing during development build launch
- fixed a bug where application would crash when returning from camera screen
- fixed issue where if camera permission was denied at the notes screen, hitting the back arrow would navigate to the task creation screen of the project tool instead

### Modified

- replaced emojis in camera screen with icons
- modified placeholder screen when camera permission denied
- cells now have a long press to delete the cell instead
- increase touchable width to encompass the entire cell, instead of just the text

### Removed

- red delete button to delete project or tasks