Making a To Do List

PROJECT PLAN

Project Plan

Summary

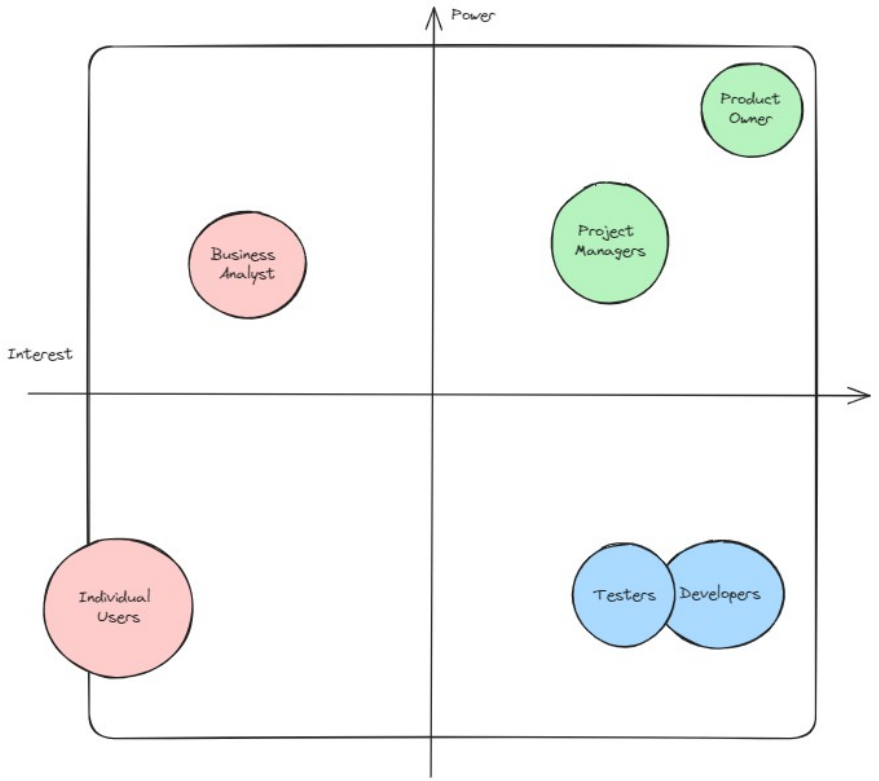
- An effective To Do List should be organised and allow the user to prioritise tasks.
- To do lists often fail when the tasks are too long and so the user will be encouraged to keep tasks short and concise. This app will include use of a timer to ensure that people are taking regular breaks. It will follow a structure like a Pomodoro Timer, so 25 mins of work, followed by 5 – 10 mins of break time. During these breaks, the app will create suggestions of how to take time away, based on mindfulness techniques and de-stressing tactics
- It should allow the user to create multiple lists, so for example, making sure there are different lists for different projects that are ongoing. You could potentially have a separate list for life admin, such as housework. Therefore, different categories could be usefil

Key Stakeholders

- On the external side of the creation of a To Do List, you have:
  i. Individual users (End-users): The person creating the list within the app and using it has an interest in the list's efficiency, and their needs must be considered during the inception of the app being made
- During the inception of the app, the key stakeholders will include:
  i. Developers: Integral in creating the software and ensuring that its functionality meets the need of the users. They may be involved in testing and debugging but a large majority of their work will be on the front-end and back-end
  ii. Product Owner: Prioritisation of project, the product owner is the face of the team within the organisation.
  iii. Project managers: Allocator of resources and ensures that budgets are kept to and that the project is completed on time
  iv. Business Analysts: Responsible for writing the accept criteria for features of the app and analysing business processes.
  v. Testers: Will take the lead on any debugging and errors. Specifically, there will be Test Managers, Test Engineers and Quality Assurance. They will work to ensure the user-journey is clear, concise and functional.

Map of Stakeholders

- Stakeholders have been mapped on the right, determined by their level of power (level of influence in the project) and interest (level of concern over the project).
- Although users are integral in terms of the final product and the functionality of the application, they are not involved in the day-to-day process of the project and have little power in the final decisions made, so they are low interest, low power.
- Product owner has highest power and interest since they are overseeing the entirety of the project and are the face of it. They will keep people in line and be important in signing off features . If the app fails, it will fall to them.
- Project manager will oversee allocation of tasks and have a hand in time management on an ongoing basis, so they have more power with more interest
- Business Analysts are the bridge between the business side and technical side of things. They will translate the needs of the business into technical specifications that the development team can understand and implement. Due to this, they have been positioned as having higher power but less interest, as they do not have a direct concern over the success of the app. As long as the dev team are aware of what they are supposed to be doing, the BA is fulfilling their job role.
- Developers and Testers have been positioned as having low power, high interest. They are involved in the day-to-day running of the project and have a lot of interest in the outcome. However, they have little power when it comes the decision-making of the application.



Meet some of the team....

James (Customer)

As the customer, the To Do List will be tailored towards his needs as a user. The success of the application is dependent on him and so initial research into potential features needs to factor in what he desires and what will be of benefit to him.

Alex (Product Owner

As the product owner, Alex will be the face of the project. Any issues will end up with him and he will be integral in organising weekly meetings, checking in on the progress of the app and acting as a “coach” of sorts for the rest of the team.

Alice (Front – End Developer)

Alice is integral in the creation of the app. Using HTML, CSS and JavaScript, she will implement the front-end of the app and ensure that the functionalities that have been decided upon are implemented, such as use of the Pomodoro timer.

Sally (Back –End Developer)

Sally is part of the back-end and she will be integral in developing the databases that will store user login information, meaning that they can access their to do lists wherever. She will use SQL, C# and other languages to ensure that the app runs smoothly.

Rob (Project Manager)

Rob is vital in setting the budget allocation for the project and is important in monitoring the progress of it, so that when it comes to any change management, they can allocate the necessary time and resources to everyone involved.

Karen (Test Engineer)

Karen is a Test Engineer and she interact with the development team to identify and resolve software bugs and errors. She can design test scenarios for usability testing and aid in preparing performance reports.

Rahul (Business Analyst)

Rahul's role as a Business Analyst is crucial in communicating the needs of the client to the development team. He must understand the client's needs and aids in monitoring the progress of the project during the development phase.

Communicating Throughout the Project

- Alex, the Product Owner, will lead a meeting once a week where all stakeholders will be involved and listen in on the progress of the To Do List application. This will begin at the conception of the project
- Rahul will be involved in meetings with the development team, where they will communicate what needs to be involved in the application and what features are to be prioritised.
- There will be regular reports submitted to the Project Manager where they can monitor the progress and report to the Product Owner of any adjustments that need to be made.

Using Customer Surveys To Determine Software Requirements For The App

- To determine what features should be used, a customer survey will be created and advertised across the company platforms, such as LinkedIn. As well as this, staff will be able to distribute the surveys to friends/family. In addition, customers will be reached through random device engagement (RDE). Using this method, surveys will be sent to a massive network of online properties, such as websites and apps.
- The questions would generally work better if they were offered as statements, so the customer could provide their answer, on a scale, rather than an open-ended question. This would likely result in more engagement. That being said, there may be instances where other question types are necessary
- As a To Do List is a bigger market, the customer surveys would help in determining any extra features that could offer a USP. It would be more difficult to determine this with a series of statements rather than questions

Example Questions

Agree Statements (answers on a scale from 1 – 5, where 1 is ‘strongly disagree’ and 5 is ‘strongly agree’

- I find a to do list useful in organising my work
- Using a to do list helps to priorities any tasks I have outstanding
- I find it easy to stick to tasks prioritised on a to do list
- I experience a lot of stress during work tasks
- I do not know how to incorporate breaks into my working day

Open-Ended Answers

- What other methods of organisation do you use to help manage your work?
- If you have ever felt burnt out because of a heavy workload, how have you overcome this?

Moscow Prioritisation Off the Back of Customer Feedback

- Must-Have
  - The ability to add, edit, complete and delete tasks
  - Prioritise tasks
- Should-Have
  - Categorise task
  - Set recurring tasks
- Could-Have
  - Attach files to task
  - Set recurring tasks
  - Share tasks with others
  - Pomodoro timer
  - Set reminders for tasks
- Won’t-Have
  - Voice commands

Software Requirements Found From Customer Research

- The most important feature will be the ability to add tasks to the list, as well as editing and deleting them.
- There will also be a function where tasks can be completed and will show in a separate task pane.
- Research also noted that people found to do lists more useful when they could categorise them, so this will be a feature as well
- The user will be able to prioritise the tasks in a specific order
- A lot of surveys indicated that they found to do lists useful in managing workloads and organisation but found it difficult to stick to them in the long term.
- Through the customer research and through the company's own research, it was found that a useful tool for better organisation is using Pomodoro Timers. This method structures work into 4 cycles of 20 – 25 mins of work, followed by 5 – 10 mins of break time. As such, the To Do List will structure tasks around this method.
- The user interface will emphasise the breaktime period via suggestions of how to best utilise the time spent away from work.

Functional v Non-Functional Requirements

Functional

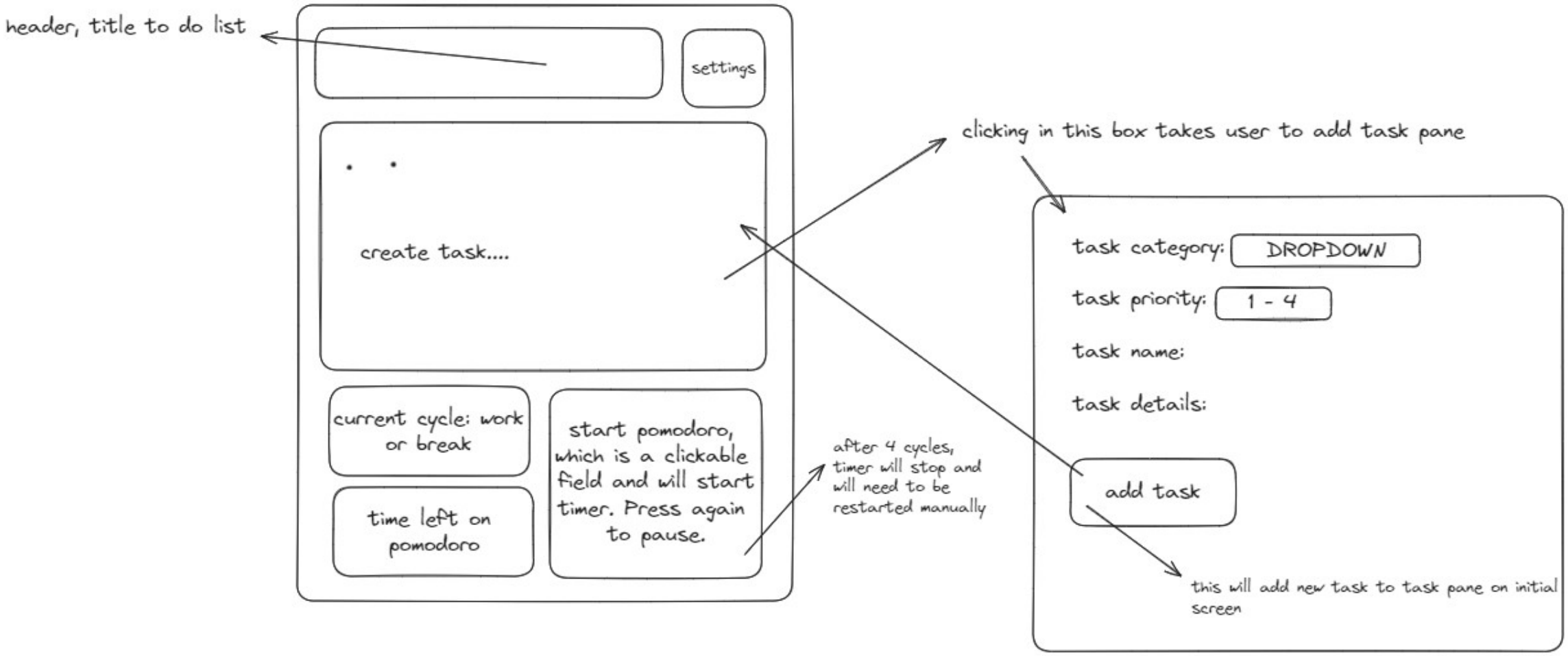
- Add, edit, delete and complete tasks
- Integration of a pomodoro timer
- User interface that emphasises how best to utilise break periods, suggesting mindfulness techniques and other ways to reset the brain before resuming work
- Categorisation of tasks
- User registration and logins

Non - Functional

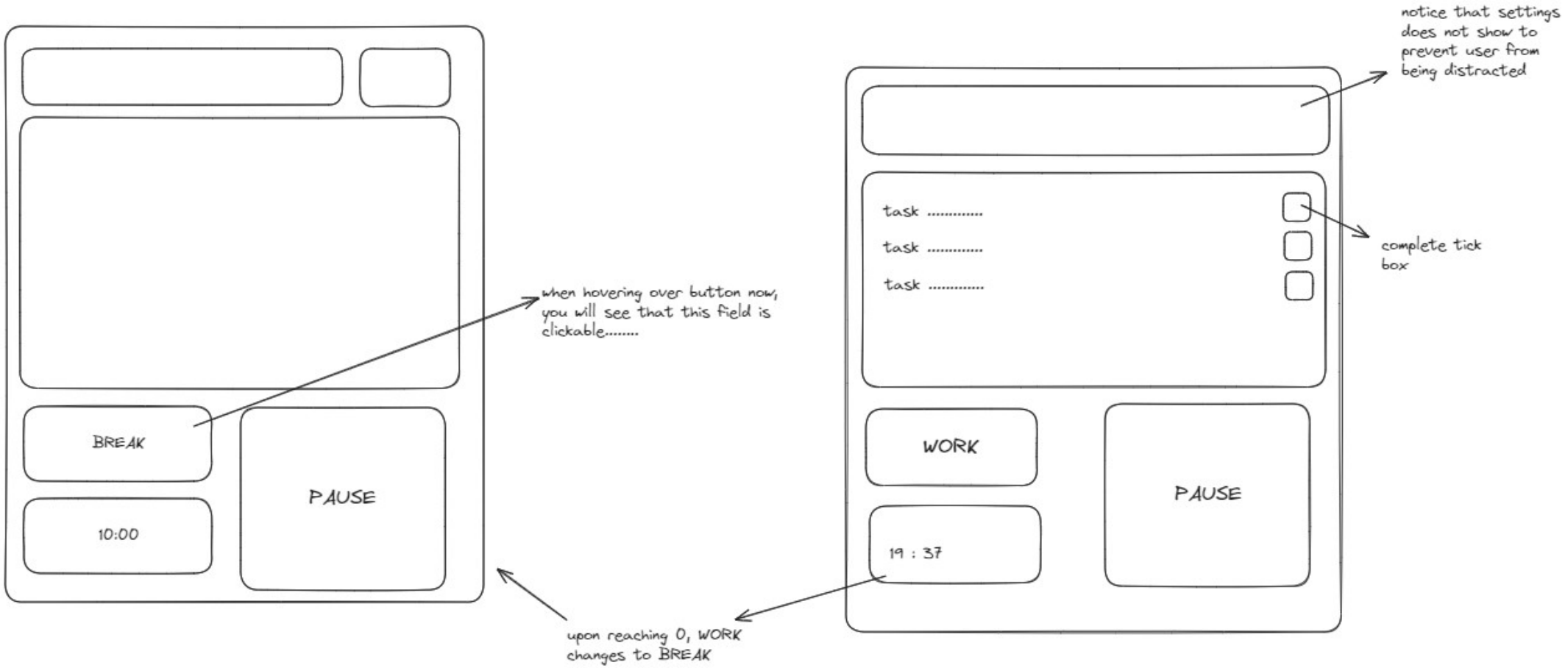
- Ability to access to do list across multiple devices, anywhere they require
- System must be easy to maintain and update
- The to do list must update within 2 seconds when a task has been marked as completed

Project Design

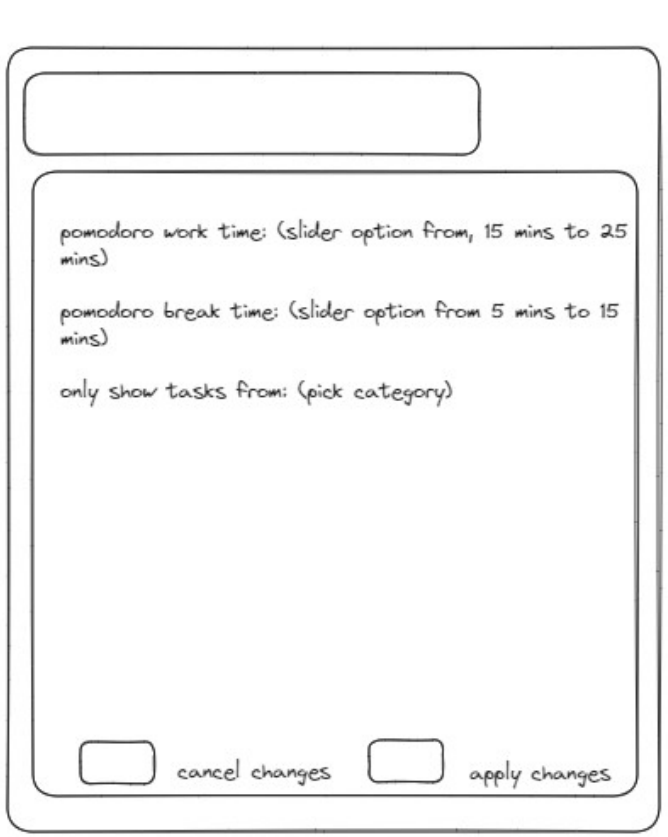
Wireframe of Main Screen, with task pane

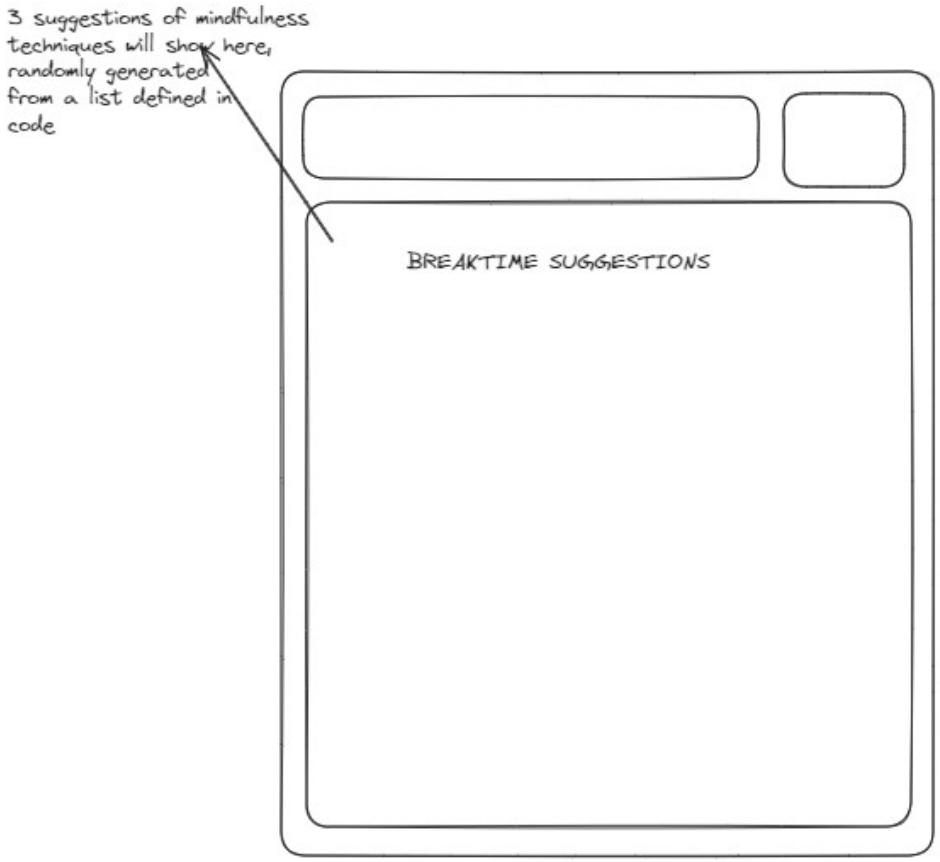


Wireframe of Updated Screen With Tasks



Wireframe of Breaktime Screen and Settings Panel





Dependencies and Requirements

- Dependences
- Task Dependencies: The development team need to determine the order in which tasks should be executed, which involves discerning how tasks are linked together.
  - For example, the actual to do list pane and pomodoro aspect can be completed separately, but the different UI designs between the break time and work time screens need to be completed after the Pomodoro is completed, as they’re dependent on the Pomodoro cycles being completed
- Internal Dependencies: The team must identify the dependencies that impact the project.
- External Dependences: Any tasks that are dependent on an external party may inform a task from being completed
  - For example, the research team will need to generate a list of mindfulness techniques/activities before they can be coded into the breaktime UI.

- Requirements
- Functional Requirements: These are to do with how the system should work and involves specific features
  - Add, edit, delete and complete tasks
  - Integration of the pomodoro timer
  - User interface that emphasises how best to utilise break periods, suggesting mindfulness techniques and other ways to reset the brain before resuming work
  - Categorisation of tasks
  - User registration and logins
- Non-Functional Requirements: These are more passive and generally linked to app performance
  - For example, how quickly the task pane loads tasks input by user. In addition, users being able to access app across different mediums

The Agile Development Approach

- This approach will be followed throughout the project so that greater flexibility is allowed for the app to be completed
- It will involve breaking down the project into smaller cycles, so essentially breaking the project down into smaller projects.
- This will allow the team to develop the To Do List app in a more iterative fashion and will be better in delivering the value to the customers

Continuous Delivery Deployment Approach

- The project will follow a continuous delivery approach.
- This is so the software release process can be more automated, from the building stage to the deployment stage.

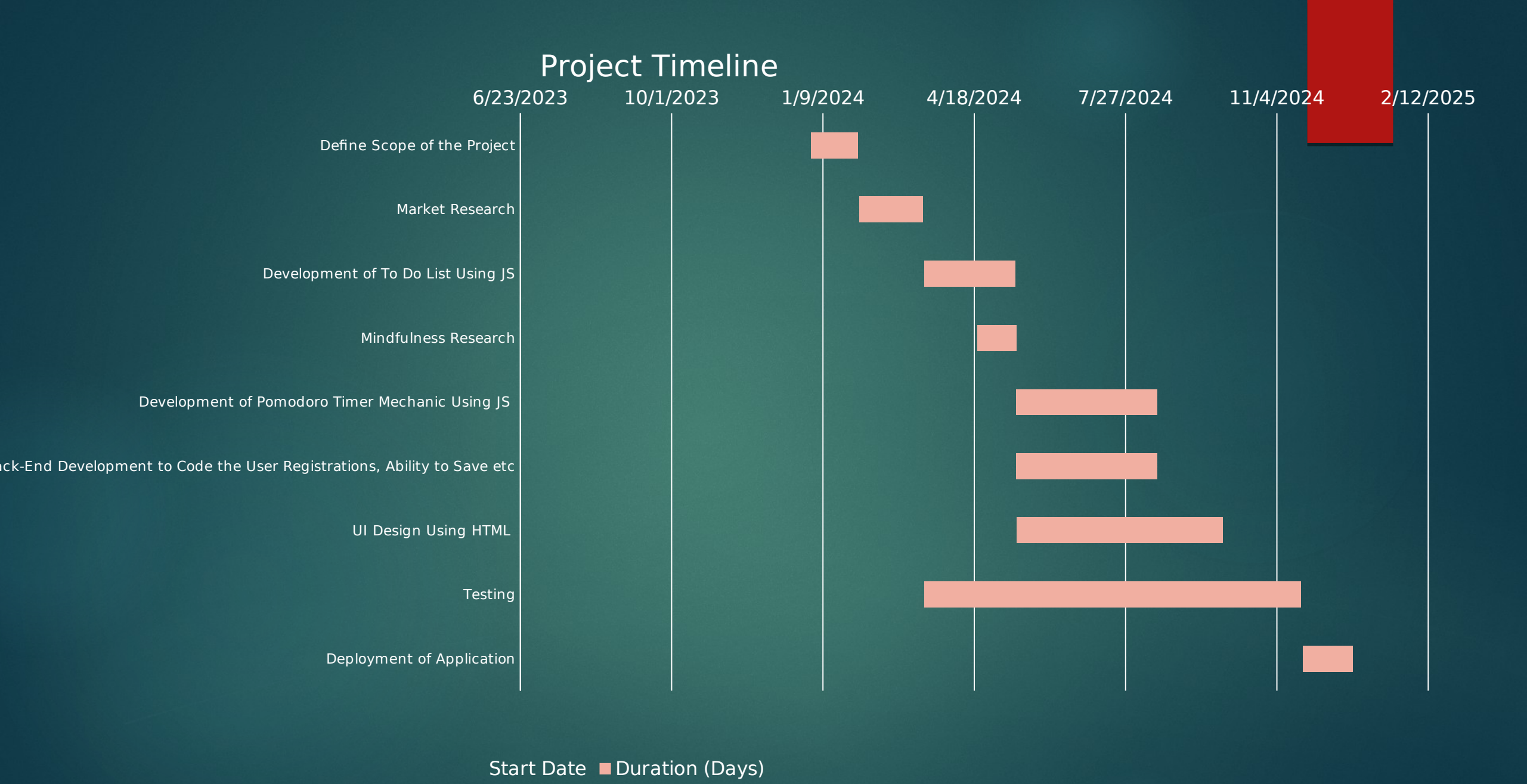
User Story – Bob Powers (CEO)

- I need a to do list that will allow me to add, amend and complete my daily tasks as required.
- I would like the app to increase my work efficiency by using other managements tactics
- I would like to be able to work on set tasks on different days, and will need to categorise them
- As I travel a lot, I want to be able to access the list wherever I am
- As I experience a lot of stress in my role, I want to be able to unwind during the moments I have as break

Project Implementation Plan

Planning Poker – Estimating The Big Tasks

- The team will utilise planning poker to estimate the length of time needed for the project.
  The individual steps to complete each part of the to do list will have been determined and the Scrum Master will allocate cards to each team member.
- For each task, each team member will allocate it a card (all cards have values from 1 – 100).
- Using this method, the Scrum Master can use these cards to discuss with the team and determine how long each stage should take



Notes on Gantt Chart

- Contingencies have been put in place for UI Design, since this is dependent on other development tasks being completed.
- Testing is put more generally but has a higher amount of time since Testers will work in conjunction with the other developers to ensure that application is working as intended. As this is a continuous approach, testers will measure and explore features as and when they are available.