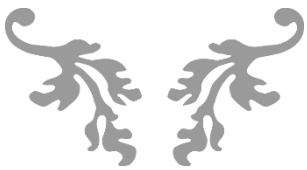



# COMPUTER-BASED PAIN DIARY

## REPORT SUBMISSION AS A REQUIREMENT FOR THE "MASTER'S PROJECT"

MSc DEGREE IN DATA ANALYSIS
AT THE SCHOOL OF COMPUTING
ROBERT GORDON UNIVERSITY
ABERDEEN, SCOTLAND



RAFAEL CASTILLO
August 2021

Supervisor DR. Roger McDermott

# Declaration

I confirm that the work contained in this MSc project report has been composed solely by myself and has not been accepted in any previous application for a degree. All sources of information have been specifically acknowledged and all verbatim extracts are distinguished by quotation marks.

Signed: Rafael Castillo  Date:17/08/2021

# Acknowledgements

I want to thank my supervisor, Dr Roger McDermott, for his support, comfort, patience, and attention he dedicated to me during my first few years at university and during this project; I respect him and am grateful to had him as a teacher at the beginning and as a supervisor at the end of my academic journey at Robert Gordon University.

I would also like to thank my partner Kamila Jekal for cheering me up during this project and motivating me to chase my goals.

And to my friend Francesco Gruosso for listening to all of my findings, plans, and conclusions of this project; having someone else talk about my project helped me to clarify and organize my thoughts.

# Abstract

This report is the continuation of the investigation of a computer-based pain diary. This project implemented the pain diary proposed during the investigation. The project uses the Waterfall method to manage the software development life cycles and feature project management techniques to keep track of progress. The web application is called "Tangible" and has been successfully design based on the functionality discussed during the investigation by creating a mock-up, flowchart, system architecture and a checklist for applying accessibility and usability techniques for the elderly. The project used the Moscow method for its requirements. During the implementation, all Must have requirements have been successfully implemented, and most should and could have as well. Some less important functionality still needs to be finished but does not significantly affect the current application. Each of these requirements has been discussed and evaluated in this report. The last chapter reflects on the project and highlights LESP issues that need to be addressed if the application wants to take further steps.

In conclusion, a chronic pain diary app has been created which allows a pain user to record essential variables to manage the pain. The application also shows insightful statistics, records of past entries and a profile screen. The application contains a novel feature that allows a user to create a spectator account that can be linked to a pain user to be spectated; the spectator can be a pain specialist or family member allowing both pain user and spectator to be on the same page. Finally, the next step of the project is to test for usability with actual chronic patients.

# Table of Contents

# Table of figures

# Chapter 1

This chapter summarizes the project investigation results and functional requirements. It will also handle project scope, management methodology, tools and resources used to design, implement, and achieve the project's objectives.

## 1.1 Background

Chronic pain affects one in every five people and is the primary cause of disability and absence in the workplace. Many problems arise for chronic pain patients and the people around them, such as family and friends. Previous research found that the most helpful method to deal with chronic pain is "pain management". Electronic pain management solutions are the most commonly used to manage pain. This project has previously analysed existing solutions and chronic pain papers and identified gaps. The project aims to design and implement a novel solution named "Tangible" that will cover some of these gaps.

Tangible will cover core functionalities of pain management solutions found on existing apps. These functionalities are compliance over paper-based solutions, valuable statistics, a web-based solution so users can store the data, access it and share it from anywhere, and record essential factors during a pain episode. Moreover, Tangible aims to fill in the following functional gaps. (1) Usability and Accessibility for the elderly, which is the most affected age group and (2) Family and friend support which is proven to enhance the quality of life of pain users.

The list below contains all the functional requirements for Tangible proposed in the investigation report.

- Must record only essential variables found in the majority of paper-based and electronic-base solutions. This will improve compliance and generate vital data for further analysis.
- Must record variables in a linear fashion where the system takes the initiative to ask the user for information, making recording a pain episode less confusing.
- Must have insightful statistic which update instantly after every pain entry. A
- Must be web-based so that information can be shared and accessed from anywhere.
- Must allow a pain user to be spectated by another member, e.g. (Pain specialist, family, friend.)
- Must include (NRS or VRS) Pain scale, i.e.( the best-proven method to measure pain intensity)
- Must incorporate accessibility measurements to allow the app to be understood and easily used by the elderly, i.e.( the most predominant age group within chronic pain population)
- Should address cultural needs by being highly customizable.
- Should have a user guide explaining the functionality of the application.
- Should have alert options so that spectator can be informed when certain factors go wrong.
- Could have reminders so that the user can schedule medicines or entries.
- Could have data export functionality allowing the user to share their pain progress with others.
- Could have all the statistical components proposed.
- Could have user log in

## 1.2 Project Scope

Tangible will be a prototype focused on functionality. Due to time constrains the following features will not be implemented. (1) The input fields in the app will not be validated. (2) Records can't be edited or deleted. (3) Only the functionality proposed will be implemented. The app is aimed at smartphones but due to MEARN being the stack known to the developer a website version of it will be created based on the smartphone design. There will be many limitations and bugs on the app that will not be addressed at this state. Not all the functionalities might be implemented.

## 1.3 Project Management Methodology

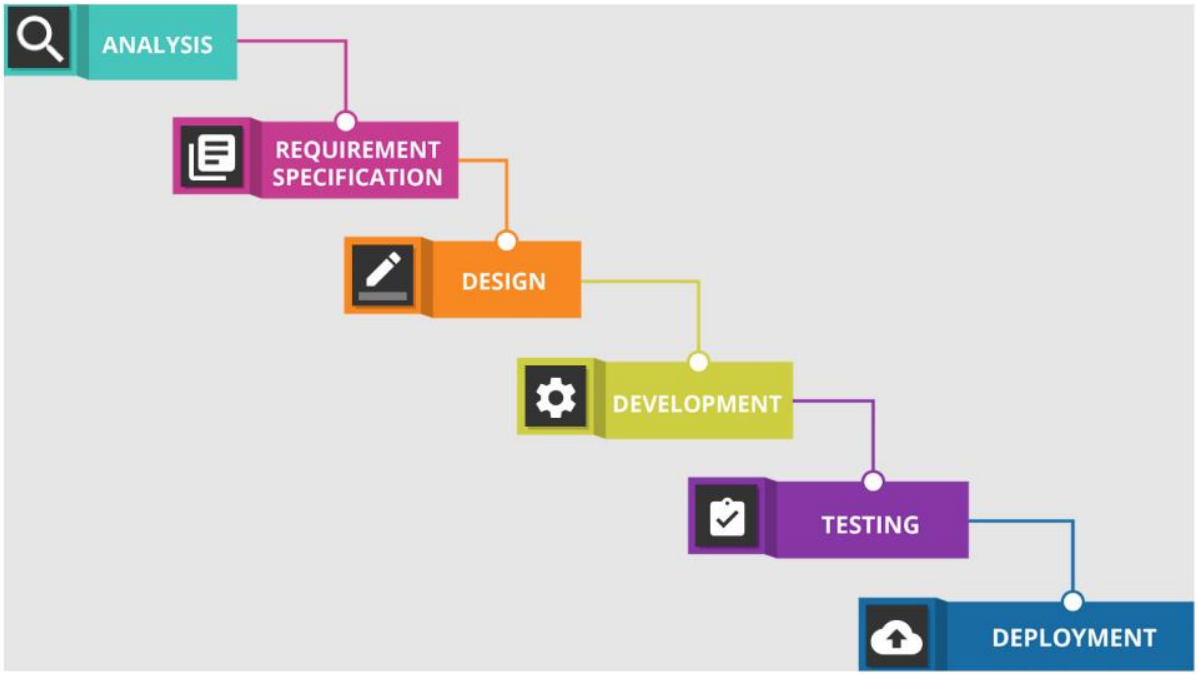



*Figure 1: Waterfall Methodology*

The methodology adopted for the software development life cycle is called "waterfall." This method relies on a sequence of steps that must be completed before moving forward to the next step; it is mainly suited for small projects when the end goal has been defined from the start, which is the case with "Tangible". The methodology focuses on a series of steps that must be followed to give the project a direction from start to finish. The

structure of this methodology is straightforward and does not require training to manage it. Figure 1 shows the steps of Waterfall in the order that has been adopted for this project.

The first two steps, "Analysis" and "Requirement Specification", have already been achieved through the project investigation before this report. The time constraint for the subsequent four phases is seven weeks. Week One will learn the tools to develop the project and set up the project management plan.
Since the project has no contact with the target market, product owner or supervisor until the end of the project and the goals are already defined, the "Design" process can be finished without big changes. This phase will be taken care of in week two.

The development phase is going to be based on the blueprint created during the design phase. The development process will last approximately three weeks. Weeks three and four will develop the front-end, and week five will take care of the back-end.

Week six will deal with testing and integration. The software will be tested for functionality; if there are significant bugs, these will be fixed.

The last week will deal with deployment. During this phase, the quality of the code comments and structure will be enhanced, and the application will possibly be deployed and hosted on a web server.

### 1.3.1 Project management Tool

The project management tool used to track the project is called "Trello". Trello offers a free service that is easy to use and highly customizable. The Trello project in Figure 2 shows the board setup. Find below an explanation for the board's setup.

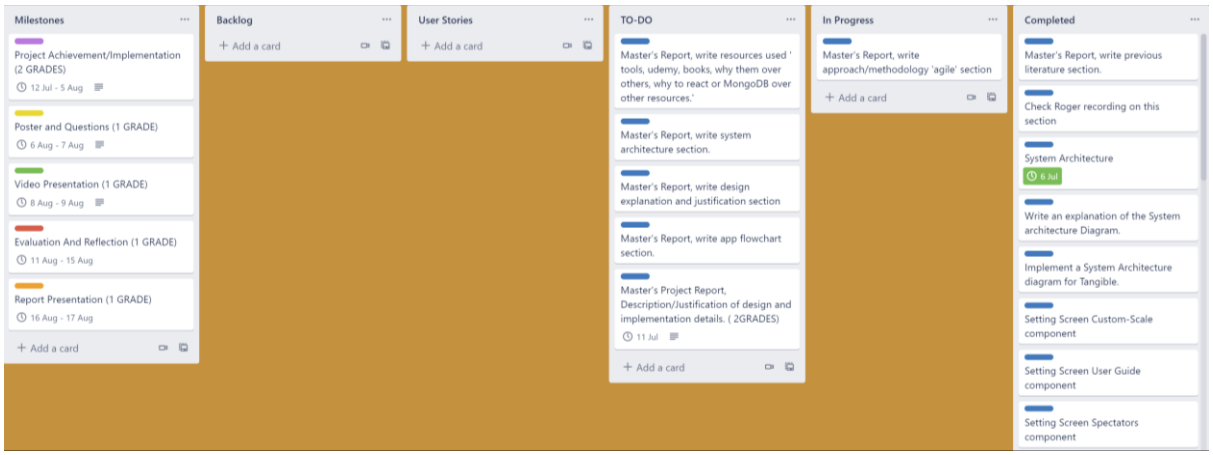



*Figure 2: Project management in Trello*

**Milestones:** The milestone section illustrates the schedule of the project. The schedule has been divided into the deliverables required for completing this master's project. Each deliverable has been timed concurrently with the development of the app. This system intends to develop the application together with the project report.

**Backlog:** There are 11 functionalities this project aims to achieve, which will be discussed in later sections. This column will help visualize and prioritize these functionalities to ensure the essential ones are being worked on first.

**To-do:** This column contains the work that needs to be done during the week.

**In-Progress:** This column contains the work that needs to be completing during the day.

**Completed:** This column contains all the report and app tasks that have been completed.

### 1.3.2 Weekly Sprints

A sprint is a timebox iteration of work. Since the length of this project is seven weeks, the length of each sprint will be a week. Every week items from the backlog will be broken down into tasks and organized by functionality. During the week, these tasks will be developed on the app. At the end of each week, there will be a retrospective to recognize obstacles that need to be addressed, and new tasks for the following week will be created. This pattern will help the project to reflect on each week of work, identify obstacles, plan and organize work. Each sprint will produce a brief report containing a summary of the retrospective; each report can be found in the Appendix A.

## 1.4 Tools and Resources Used

This section is going to talk about the different tools and resources used to design and develop Tangible.

**Figma**: This tool is a vector graphics editor used for creating software graphical interfaces. The tool has been used to create the interface for Tangible and some of the infographics shown in this report. There are alternative options to Figma, such as AdobeXD, but Figma has been the tool of preference for this project due to the familiarity with the tool.

**Udemy**: This is the online course provider used to learn the full-stack web development skills needed to complete this project. Udemy offers affordable, good quality courses and has been used because an account with the needed courses was provided. The courses taken for this project have been listed and referenced below.
React – The complete guide (incl hooks, react router, redux) ("React 16: The Complete Course (incl. React Router 4 & Redux)," n.d.)
NodeJS – The complete guide (MVC, REST APIs, GraphQL, Deno) by ("React 16: The Complete Course (incl. React Router 4 & Redux)," n.d.)

**Web Stack**: The web stack selected to develop the application is FERN, which comprises Firebase, Express, React, and Node.js. Other stacks include LAMP or MEAN, which are good stacks to build this web solution, but FERN has been selected to be familiar with the tool.

**Refactoring UI**: This book contains many software designs, accessibility and usability methods to create a good quality website. The book's writer is a dedicated web developer who also designed and built their own CSS framework. This book was selected due to the high-quality advice in UX-UI and the book's accessibility.

**CSS Framework**: The CSS framework selected for this project is "Material UI". There are other Interfaces such as Bootstraps, Cirrus, Bulman, but Material UI has been selected due to its compatibility with React. The framework has been designed by Google, making it a reliable tool. The documentation of the framework is of high quality. The framework offers a familiar look mainly observed in google websites and android smartphone application; since this project aims to deploy the solution on a future as a phone app is convenient for the web application to adopt such familiar design.

## 1.5 Report Structure

**Chapter 2**

This chapter will showcase, explain, and justify the system architecture, design, accessibility and usability techniques used to build Tangible.

**Chapter 3**

This chapter evaluates the quality and effectiveness of implementing the objectives discussed in section 1.3 of this report. The chapter will also discuss improvements for future work.

**Chapter 4**

This chapter will reflect on the project management techniques and will also highlight LESP issues related to the project. Lastly it contains the conclusions and future work for Tangible.

# Chapter 2

This chapter will showcase, explain, and justify the system architecture, design, accessibility, and usability techniques used to build Tangible.

## 2.1 System Architecture

A system architecture is used to understand different aspects of a software application. The subsections below will explain the different aspects displayed in the system architecture shown in Figure 3

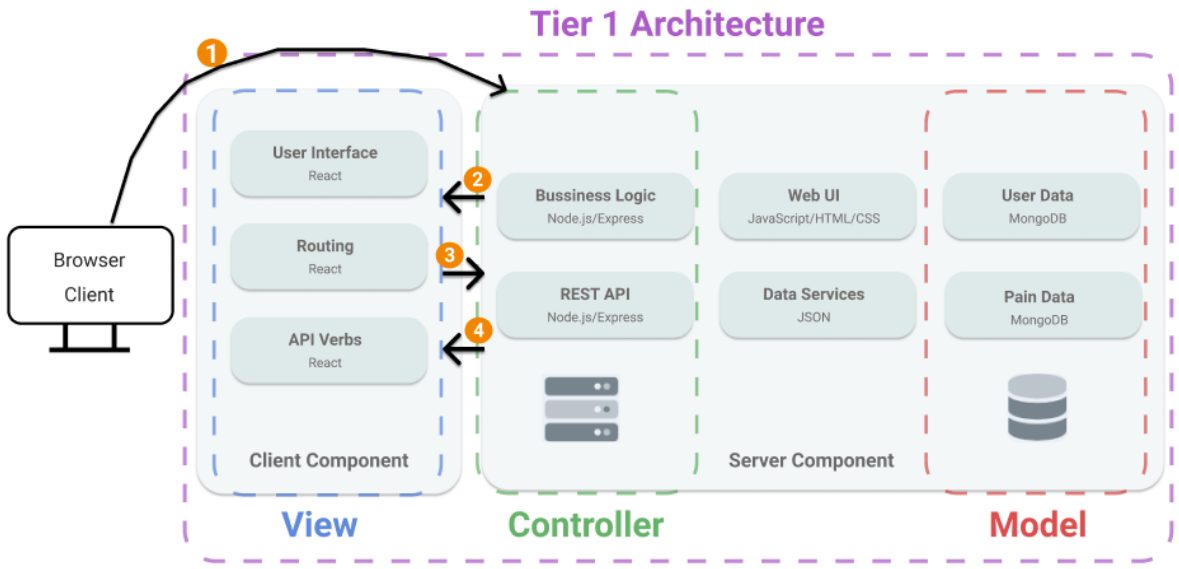



*Figure 3: 1-Tier SPA MVC Monolithic Architecture*

### 2.1.1 Monolithic Architecture

The system architecture implemented for Tangible is “The monolithic” architecture due to the advantages of debugging, testing and deployment. Alternatively, if the application grew more complex and popular, adopting a microservice architecture could improve scalability, security, and a clearer understanding of the functional structure.

### 2.1.2 Software Architecture Pattern

Tangible implements MVC, a software architecture pattern that divides the functionality of the application into three components (1) "View" represents the UI components a user can interact with, (2)"Model" represents how data is stored in databases and (3)"Controller" which connects the model and view. Because MVC divides the functionality, i.e. (presentation and business logic), it makes it easier to design, build, debug, maintain, modify and test the code in the app.

### 2.2.3 Tier Architecture

Tangible has been built on a 1-tier architecture system. This system uses a single server and database all in one place, allowing the app to be highly connected; this means no network latency and highly available data making it simple to develop and test the app. Alternatively, once the application is officially deployed, a higher tier architecture will be needed due to tier-1 being bad for scalability, portability, maintenance, safety, and reliability.

### 2.2.4 Web Application Architecture

A web application architecture is a pattern of interaction between client and server-side components. Tangible uses a Single-Page Application (SPA) pattern. This approach results in an uninterrupted, intuitive and interactive user experience by loading the entire web UI into the client browser and updating only the individual UI components the user interacts with, therefore limiting page loads and achieving fast response times. This approach is also easier to build, debug, deploy and has a smooth transition to mobile development. The only downside to SPA is the long initial wait time when loading the entire web UI. The flow of a SPA has been labelled in Figure 3 by the four steps using numbers and arrows. Step one is the initial client HTTP request when entering the website for the first time. The second step is the initial HTTP response containing and loading the complete website in the client browser. Steps three and four are the API jason requests performed by the user to change or update specific parts of the website.

## 2.2 User Interface functionality base Design

This section is going to show, explain and justify the design chosen for Tangible.

### 2.2.1 Application flow

The application has three main screens which contain the functionality of the application. Each of these screens is interconnected by a menu where the pain user can choose between entries, stats, and profile screens. Figure 4 shows the relation and application flow between the three screens.

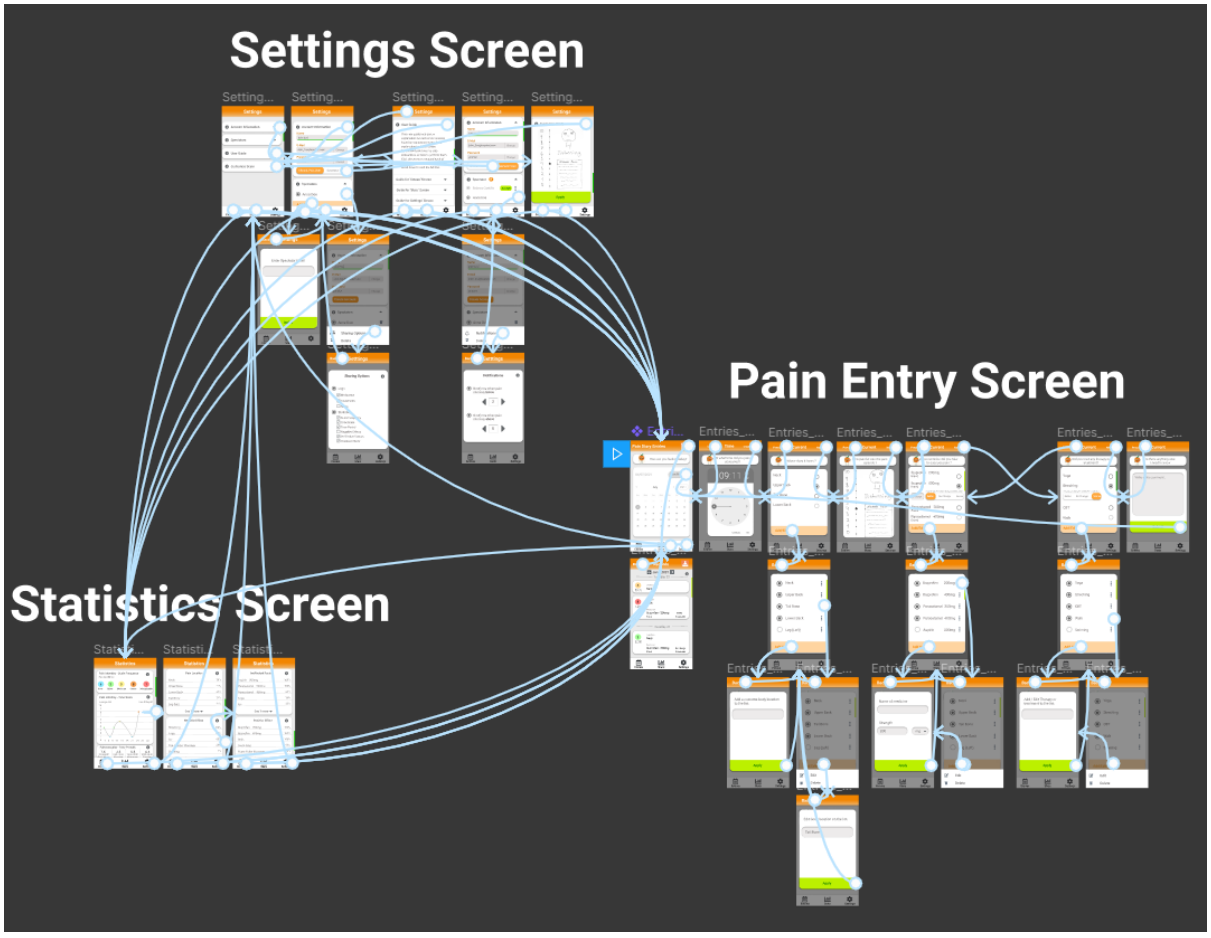



*Figure 4: Application flowchart/Interactive Mock-up*

**Pain Entry Screen:** This screen counts with the records screen and seven sub-screens for recording entries. Both of these functionalities have the menu bar enable to change to the other two screens. The flow shows that once a new entry has been started, the menu bar becomes unavailable and forces the user into a linear interaction where the system takes the initiative to ask for the correct user input, making this approach suer friendly.

**Statistics screen:** This screen can be scrolled down to display the different charts that form it, seven in total. The menu bar on this screen is enabled at all times.

**Settings screen:** This counts with one screen, containing an accordion with four sub-categories (account info, spectator, user guide, scale customization). This screen can swap to any of the other screens unless the user is currently customizing a functionality; during this interaction, the main menu bar is disabled, and the user must cancel or apply the functionality before swapping to other screens is enabled, this behaviour prevents bugs.

Figma also offers an interactive experience to understand the flow of the application better. An interactive mock-up has been created for this application Figure 4

## 2.2.2 Entries Screen

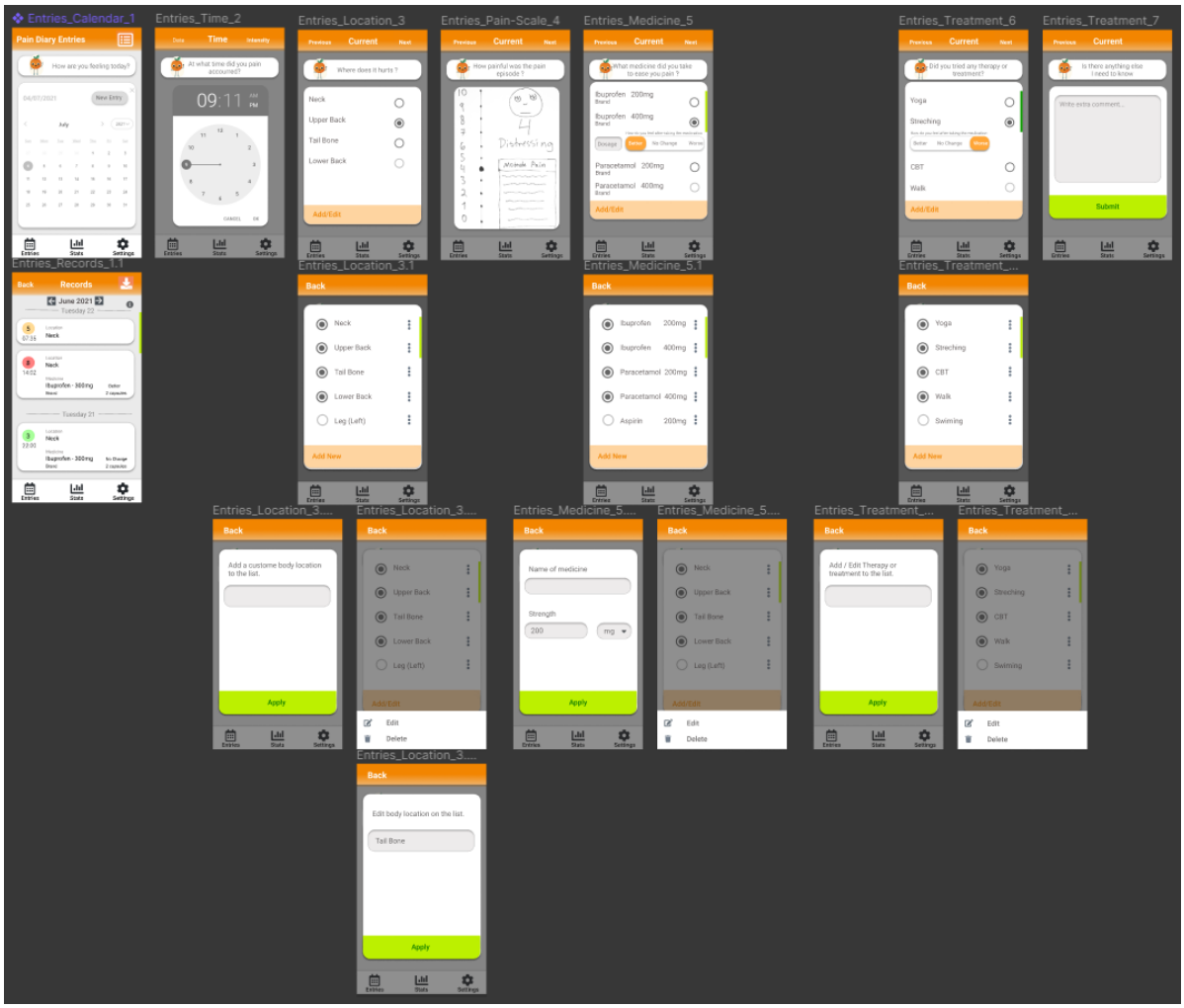



*Figure 5: Application Mock-up - Entries Screen*

The design for the entries screen is shown in Figure 5. This screen allows the pain user to review their records and create new pain entries. The user can customize most variables and move back and forth between the variables being recorded. Use instruction and explanation for each screen has been provided below.

### 2.2.2.1 Logs

In order to access records, the user must go to the "Entries screen" and press the calendar icon in the up-right corner.

Once in the "Records" screen, the user can scroll down to see past logs corresponding to the current month. If the user wants to check a previous month, he can interact with the

date-changer at the top of the screen, which allows him to select previous months by clicking on the arrows. Additionally, the user can press on the informational tooltip to obtain instructions on operating this screen.
The user can download logs as a CSV file by clicking on the download icon on the top-right corner.

#### 2.3.3.2 Starting an entry

Press the "New Entry" button at the top right of the calendar on the 'Entries' screen to start a new entry. Tangible asks six questions about the pain episode. The user can skip or go to the next question by selecting the next section on the top-right corner of the screen; subsequently, the user can go back to the previous question by pressing the previous section at the top-left side of the screen. The six questions are discussed below.

**Choosing Date**

On the calendar component, the user can select the date number, and at the top of the calendar, the user can select the month by pressing the arrows or select the year by pressing the drop-down menu on the up-right corner of the calendar.

**Time of the episode?**

The user can select the time when the pain episode occurred by pressing on the time-picker. The time-picker asks for the hour of the day first and then minutes. The user can choose to change the hour or minutes by pressing down on the desired section. Additionally, the user can also choose between AM or PM by pressing down on the option within the clock.

**Location of the pain?**

Initially, this screen will be empty. The user will have to add the locations in the body where the pain is felt.

Once Locations have been added, the user can select the location on the body where it feels pain by pressing the empty circle-shaped button.

The user can access more options by pressing the Add/Edit button at the bottom. The user can scroll down on this screen and enable or disable multiple body locations by pressing the circle-shaped button. Body locations that have been enabled will show on the main screen.

**Editing / Deleting / Adding body locations**

The user can also press on the vertical three-dot icon to delete or edit a body location or add a new one by selecting the "Add New" button at the bottom.

**Pain Intensity**

The user can use the vertical scale to select the intensity of the pain between 0-10. 0 stands for "no pain", and 10 stands for the "worst pain imaginable". Every time the user picks a different level of intensity, a description corresponding to that intensity will be shown. Descriptions are meant to help the user choose the accurate intensity of the pain.

**Medicine Taken**

Initially, this screen will be empty. The user has to enter the medicines taken to mitigate the pain.

Once medicines have been added, the user can select the medicine taken to mitigate the pain by pressing the empty circle-shaped button. When a medicine is selected two new attributes will be displayed. The first attribute asks for the dose taken, and the second attribute whether that medicine was effective, ineffective or counter-effective.

The user can access more options by pressing the Add/Edit button at the bottom. The user can scroll down on this screen and choose to enable or disable multiple medicines by pressing the circle-shaped button. Medicines that have been enabled will show on the main screen.

**Editing / Deleting / Adding Medicines**

The user can also press on the vertical three-dot icon to delete or edit a medicine or add a new one by selecting the "Add New" button at the bottom.

**Treatment or Therapy taken**

Initially, this screen will be empty. The user has to enter the method taken to mitigate the pain.

Once methods have been added, the user can select the approach taken to mitigate the pain by pressing the empty circle-shaped button. When a method is selected a new attribute will be displayed to choose whether that method taken was effective, ineffective or counter-effective.

The user can access more options by pressing the Add/Edit button at the bottom. The user can scroll down on this screen and choose to enable or disable multiple methods by pressing the circle-shaped button. Methods that have been enabled will show on the main screen.

**Editing / Deleting / Adding methods**

The user can also press on the vertical three-dot icon to delete or edit a method or add a new one by selecting the "Add New" button at the bottom.

**EXTRA COMMENTS**

The user can write additional comments about the pain episode. Once the user has finished writing, he can press the submit button at the bottom to conclude the pain entry form currently recorded.

### 2.2.3 Stats Screen

The design for the stats screen is shown in Figure 6. This screen allows the pain user to visualize their pain records through insightful charts to understand the pain better and spot good, bad habits. Seven charts display information about the pain. Each chart has a title explaining its purpose and an informational tooltip explaining how to understand it. Explanations of these charts can be found below.

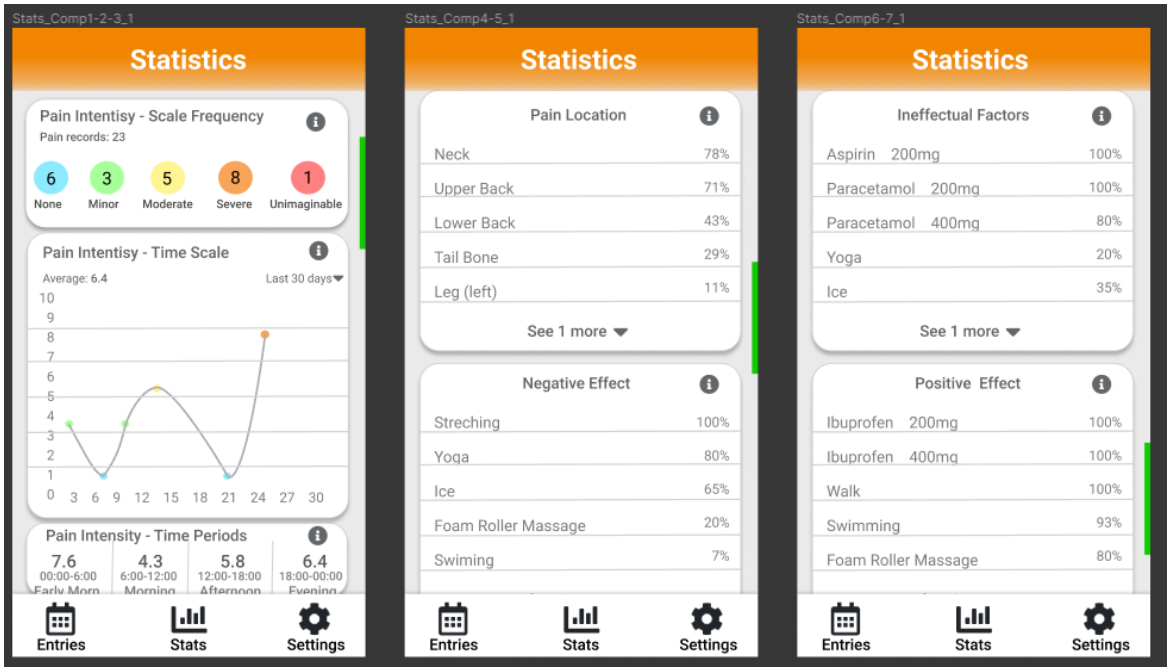



*Figure 6: Application Mock-up - Statistics Screen*

#### 2.2.3.1 Pain Intensity "All Records"

This chart shows the total amount of times a user has recorded a pain episode. The intensity level recorded during an episode has been distributed across five different pain categories.

This chart can keep a user accountable for the number of entries recorded. It also helps to have a general understanding and classify their overall pain severity.

#### 2.2.3.2 Pain Intensity "Time Graph"

This interactive chart displays pain intensity on the vertical line and the time range on the horizontal line. The user can change the time range displayed by pressing on the arrow pointing down located at the top-right side of the chart.

The chart shows the average pain level across the selected time range. The graph also shows a line displaying the pain level episode across the selected time range. The user can compare the average pain level or line graph across time, allowing him to understand their pain flow.

#### 2.2.3.3 Average Pain Intensity "Time Periods"

This chart shows the average pain level across four-time categories. Each category represents a range of times in a day.

This chart helps to understand at what time throughout the day the pain is most severe. This user can use this to find trends, e.g.(when the pain was most or less severe) by investigating the habits during those periods, enabling them to adopt better or discontinue poor habits.

#### 2.2.3.4 Pain Location

This chart shows how many times a body location has been selected when filling a pain entry form. It also shows the most prevalent body location affected by the pain.

This chart helps the user to understand the parts of their body which are commonly affected by the pain, enabling them to choose the proper medication and treatment to mitigate the pain.

#### 2.2.3.5 Negative Effect

This chart shows the number of times a pain mitigation method was classified as "worse" when filling a pain entry form. It also shows which methods have the most harmful results when treating the pain.

This chart can help the user to keep accountability of how many times a method has been utilized. It also helps the user to recognize which methods are not suitable for them and must be discontinued.

### 2.2.3.6 Positive Effect

This chart shows the number of times a pain mitigation method was classified as "Better" when filling a pain entry form. It also shows which methods have the most beneficial results when treating the pain.

This chart can help the user to keep accountability of how many times a method has been utilized. It also helps the user recognize which methods have worked in the past and should not be abandoned.

### 2.2.3.7 Negative Effect

This chart shows the number of times a pain mitigation method was classified as "No change" when filling a pain entry form. It also shows which methods have no effects when treating the pain.

This chart can help the user to keep accountability of how many times a method has been utilized. It also helps the user to recognize which methods are ineffective for them could be discontinued.

### 2.2.4 Profile Screen

The user can record or access personal information. The user can also manage spectators, customize the app and read the user guide. There are four sections in the profile screen which are discussed below.

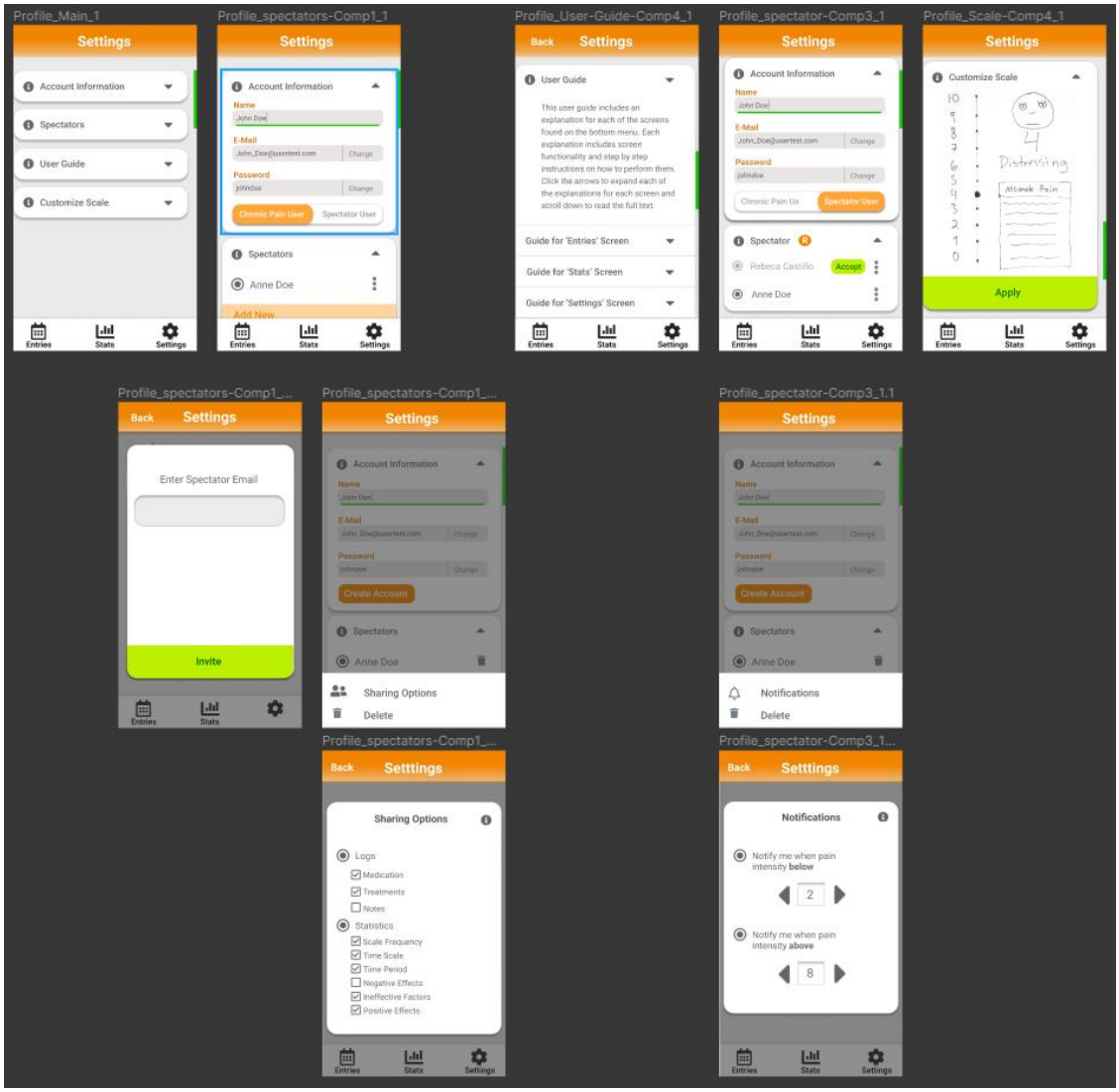



*Figure 7: Application Mock-up - Profile Screen*

#### 2.2.4.1 Account Information

This section manages the information in the account. The user can see account type or choose a name, email and password associated with the account. This information is used when connecting with spectators and storing pain records in the cloud.

**Creating an account**

The user will be asked to create an account when using the application for the first time. The user must choose between a chronic pain or a spectator user.

**Change email and Password**

The user can click on the "Change" button found on the email or password line in the account information section. A new window will pop up where the user will have to enter the previous email/password and the new email/password the user desires to change and click on the "change" button.

### 2.2.4.2 Spectators

This section will look differently depending on the user being a spectator or a chronic pain user. The section allows the spectator and pain user to manage connections between them.

**Spectators for "Chronic Pain User"**

The user can change permissions for spectators by pressing the circle-shape button to enable or disable the spectator's ability to see their information.

**Adding a new spectator**

The user can add spectators by clicking on the "Add New" button and enter the email associated with the Tangible account of the spectator they wish to connect.

**Removing a spectator**

The user can remove a spectator in the spectator's section by pressing the vertical three-dot icon and then pressing on the option "Delete".

**What information can a spectator see?**

The pain user can select the information that a spectator is allowed to see. The user can exclude different factors, e.g. (medicine, negative effect chart). To choose what information a spectator can see, the user must go to the spectator's section and press on the vertical three-dot icon on the spectator’s name he desires to customize; the user can

press the "Sharing Options" button and choose which factors that spectator is allowed to see by checking or unchecking the boxes in that window.

**Spectators for "Spectator User"**

The user can select which pain users he wishes to spectate by pressing the circle-shape button to enable or disable the pain user hence stopping all notifications associated with that pain user. The spectator can also see requests sent by a pain user in order to be spectated. The spectator can also manage other pain user options discussed below.

**Accepting/Deleting Request**

When a pain user sends a spectator request, a tooltip showing requests will appear in the spectator section. To accept a request, the user can press the "accept" button or press the vertical three-dot icon and choose to delete the request.

**Notify the spectator when the pain user has an episode.**

By selecting the three-dot icon in an existing pain user on the spectator section, the spectator can select the notifications button and choose a threshold to be notified when the pain user has a pain episode above or below a chosen threshold.

### 2.2.4.3 Customize Pain Scale

In the customize scale section, the user can select a number from the pain scale and edit the description on the right. The user must press the "Apply" button for the changes to take effect.

### 2.2.4.4 User Guide

This user guide includes an explanation for each of the screens found on the bottom menu. Each explanation includes screen functionality and step by step instructions on how to perform them. Click the arrows to expand each of the explanations for each screen and scroll down to read the full text.

## 2.3 User Interface Accessabitiy and Usability

Usability and Accessibility are related features of a web solution that works for most of the target users. Usability focuses on designing effective, efficient and satisfying interfaces to enhance user experience. Accessibility addresses people disabilities, e.g.(colour blindness, old age); these concepts make a tailored solution for those user groups. As a result, people with disabilities can equally perceive, use and understand the application as much as people without disabilities.

There are many user groups to consider when designing an application for managing chronic pain. For this initial state, the majority user group, "the elderly," has been selected to address usability and Accessibility based on their needs. The usability and accessibility features implemented for Tangible are discussed below.

The following steps have been taken to help address motor and vision issues found in the elderly group. The information has been gathered from the book previously mentioned ins ection 1.6 if this report.

### 2.3.1 Font

Typography influences how the application feels. Fonts with round ages make the app look playful were as sharp edges look elegant.

The font of choice for Tangible is "Roboto". Google has specifically designed this font to balance content density and reading comfort on a screen. The font renders in most web browsers and is used by many Google products, making it popular and familiar. Each character in Roboto has been crafted to stand apart and are easier to read for visually impaired users. The font is also compatible with screen readers.

The font size selected for Tangible is 16px on the phone and 19px on tablet/desktop. These are the font size recommended addressing accessibility concerns.

### 2.3.2 Constrast

The contrast between the font and background are also essential when addressing accessibility. The Web Content Accessibility Guidelines recommend that standard text (under ~18px) has a contrast ratio of at least 4.5:1, and that larger text has a contrast ratio of at least 3:1.

### 2.3.3 Paragraph

The length of the character in each paragraph line should be 50-60 characters long. Short lines are easier to read for all users, mainly for users with visual impairments. The space between the lines in the paragraph must 30%, or more over the font size used. In the case of Tangible, this should be 21px for phones and 25px for other screens. Justification and Hyphenation of text will also be avoided not to make the content harder to read, especially for people with visual impairments.

### 2.3.4 Other

- Typing has been kept to a minimum, and most user input is through selection.
- The interaction with the device has been kept short and straightforward; recording a pain episode should not take longer than a minute.
- Text will not be overlayed on top of images or graphics unless it is contrasted.
- The use of multiple fonts has been avoided.
- Material UI provides a font scale for consistency.
- Red and green have been avoided as font colours due to these colours being the hardest to read for colour-blind users.
- A friendly tone has been used as a language for the application, which makes it more social.
- The border of UI components has a large border radius for a fun, playful feel.
- The contrast between components will create depth and allow the user to differentiate the interface better.
- Buttons or other input targeted components must be of the minimum size of 44 by 44 CSS pixels.
- Headings must use the correct HTML tags; this enhances accessibility helping all users understand the content, and search engines use them to understand a site's content.

- The app has been named "Tangible" after the idea that "pain is subjective and intangible, but this application can make it tangible through the pain management techniques implemented".
- The feel and look of the app aim to give chronic pain users a healthy, positive, fun, supportive and interactive environment. The logo for the application is a fruit, "tangerine"; this is because both words "Tangerine and Tangible" can be easily associated by sound and letters hence easily remembered. Subsequently, fruits relate to concepts such as healthy and positive, which the app portraits.

### 2.3.5 Colors

The colours found in tangerine have been selected for the application. The tangerine is the primary Colour of the app. This Colour is linked to motivation, lends a positive attitude, and general enthusiasm for life. It is excellent for bringing comfort in tough times, creating a sense of fun or freedom, and not allowing you to sink into grief or disappointment.

The Colour green has been selected as a secondary colour. This Colour is associated with vegetation, harmony and peace. Green is portrayed as a calming shade away from the stresses of everyday life.

### 2.3.6 Other change name

The following steps have been taken to help address Memory and concentration issues found in the elderly group.

- The application includes informational tooltips and a user guide for step by step instructions on how to use it.
- The pain entries are recorded linearly where the system takes the initiative of asking for the information the user must input.
- Scrolling has been avoided in most cases or kept short.
- Symbols have been paired with text in most cases.
- Actions have been adequately labelled, and feedback, when the action is taken, has been provided.
- The app header menu on the top informs the user of its location and the previous location he came from.

# Chapter 3

The following chapter will evaluate the quality and effectiveness of the implementation of objectives discussed in section 1.3 of this report. The chapter will also discuss improvements for future work. Each component will be judged under the following criteria. (1) Was the functionality fully implemented? (2) Does it meet the project specifications? (3) Were there any obstacles? If yes, how was it solved? (4) Are there any bugs? (5) Any ideas for improvement? (6) general difficulty for the task?
Lastly, there will be a discussion about the three different types of testing implemented to address the functionality, interface, and security features.

## 3.1 Must record essential variables. Must record variables in a linear fashion. Must include (NRS or VRS) Pain scale. Should be highly customizable.

The app must record the essential variables proposed during the investigation report (date and time, pain location, pain intensity, medicine, treatment, additional comments).

### 3.1.1 Date and Time

Two buttons have been created, “Back” and “Next”, forcing a linear behaviour when recording a pain episode. The system takes the initiative to ask the user for the information required by placing a question/text on a UI card.

**Meets project specification / Functionality**

The component fully meets the project specification as it allows the user to record the date and time of the pain episode successfully.
The date and time functionality has been fully implemented.

**Obstacles**

- Building a calendar and time picker was too time-consuming to do from the start, proving to be an obstacle. Since React has the luxury of having a big community that provides free components, research for a calendar and time picker components was done. The current calendar used by tangible has been obtained from the following source: (Freoksenet, Anoobis, & Wojtekmaj, 2021).

**Bugs**

- There are currently no bugs detected with this functionality.

**Improvements**

- The functionality could be improved by forcing the user to pick past or current dates only.
- The time picker needs to be made more accessible because managing a dropdown menu can results difficult for the elderly.

**Difficulty**

Having to deal with third-party components and the date format proved to be a lengthy and challenging task. Careful study of documentation and extensive research on manipulating date format was necessary.

### 3.1.2 Pain Location, Medication, and Treatment

These three components are similar hence will be reviewed together.

**Meets project specification/ Functionality**

The components fully meet the project specification as it allows the user to customize the components fully and successfully record pain location, medication and treatments exerted and their effectiveness to mitigate the pain episode.

The pain location, medication and treatment recording functionalities have been fully implemented.

**Obstacles**

- The biggest obstacle when building this functionality was the UI, as it required learning how to use the Material UI framework, which proves to be a lengthy and challenging task. Extensive reading and analysis of examples and documentation provided by the official Material UI page were needed to implement the components. This information can be found at the following source: (Google, n.d.).
- Another obstacle was figuring out how to make the data persistent; all actions, including enabling/disabling, adding, editing and deleting options, send a fetch request with the current settings, this information is then fetched back when the component re-renders. Because these components get rendered when the user selects Next or Back, they lose their previous state, causing problems; to solve this, a state must be created on their parent component, which does not re-renders

hence it can store the options previously selected by the user without causing problems.

**Bugs**

- Currently, no bugs have been identified with this functionality.

**Improvements**

- These components can be improved by adding input validation and allowing their menus to scroll when the list of options grows large.
- Every time one of these components renders, it fetches the current options to the user; by fetching this data on user login and managing their state on a parent component as discussed in the previous point, the application can obtain a significant optimization by fetching data only once.
- Confirmation when deleting an element needs to be implemented.
- These three individual components can be re-worked into one reusable component, reducing the amount of code within the app.

**Difficulty**

This task was relatively easy to implement but required extensive time and understanding of how to move data and functions between React components, making the task daunting and lengthy.

### 3.1.3 Pain Scale

**Meets project specification/ Functionality**

The component fully meets the project specification as it allows the user to successfully record pain intensity by using VRS and NRS rating scales, as is the case with the NHS sample scale in the Appendix B

Pain scale Functionality has been successfully implemented

**Obstacles**

- The only obstacle was the understanding and proper implementation of the components from the Material UI framework

**Bugs**

- No bugs detected

**Improvements**

- Improvements can be made by improving the dynamic text's grammar and credibility and having it approved by pain specialists.

**Difficulty**

Overall, this component was straightforward to implement.

### 3.1.4 Extra Comments

**Meets project specification/ Functionality**

The component fully meets the project specification as it allows the user to record additional comments successfully.

Extra comments functionality has been successfully implemented.

**Obstacles**

- There were no obstacles when implementing this component

**Bugs**

- No bugs detected

**Improvements**

- Improvements could be made by showing a summary screen of the recorded data after submission
- Modal showing feedback to the user that the pain episode was successfully recorded.

**Difficulty**

This component was straightforward to implement.

## 3.2 Must have insightful statistics. Could have all the statistical components proposed.

The investigation report proposed a total of 7 charts. Due to time constraints, 6 of these charts were a Must have and the last chart would be optionally added if there was enough time. All must chart shave been successfully implemented and meet the project specification. The optional graph has not been implemented due to a shortage of time.

### 3.2.1 Scale Frequency

**Meets project specification/Functionality**

The component fully meets the project specification as it allows the user to view the total amount of recordings and to which pain category they belong.

Time-frequency functionality has been successfully implemented.

**Obstacles**

- No obstacles emerged

**Bugs**

- No bugs detected

**Improvements**

- This component could be improved by making it dynamic for smaller screens.

**Difficulty**

The component was straightforward to implement.

### 3.2.2 Time Periods

**Meets project specification/Functionality**

The component fully meets the project specification as it allows the user to view the intensity of its pain distributed by different time periods. In addition, the number of occurrences was also added to the component to add context to the pain average.

Pain periods functionality has been successfully implemented.

**Obstacles**

- No obstacles emerged

**Bugs**

- No bugs detected

**Improvements**

- This component could be improved by making it dynamic for smaller screens.

**Difficulty**

The development of the component was straightforward.

### 3.2.3 Location and Factor graphs

**Meets project specification/Functionality**

The component fully meets the project specification as it allows the user to view the pain location frequency and the positive, negative and ineffectual pain management techniques used to handle the pain.

Graph functionality has been successfully implemented.

**Obstacles**

- The library used to apply the components required to carefully review and understand the documentation and examples provided by Recharts from the source: (Recharts Group, n.d.).
- The algorithms needed to manipulate the data and do the statistical calculations were complex; this had to be worked in its own file and divided into helper functions to assist the primary method; code in this section had to be adequately documented.

**Bugs**

- No bugs detected

**Improvements**

- Items added on the graphs stack on top of each other, making it unfit for many locations or factors. This feature can be improved by adding a scroll functionality when the list gets large and allowing the user to delete or archive items in the graphs.

**Difficulty**

This section proves to be daunting to implement.

## 3.3 Must be web-based.

**Meets project specification/Functionality**

The app is hosted on the web by GitHub Page; this option has been selected due to familiarity and because it offers a free service. The user information is stored in firebase; this option has been selected due to familiarity, helpful API, and simplicity of use. These two features make the project web-based, meeting the project specifications.

Web-based functionality has been mostly implemented; A web host which provides a single-page application service needs to be adopted for this requirement to be fully met.

**Obstacles**

- In order to use FireBase API, extensive research and documentation were needed.

Bugs

- When refreshing/reloading the website after the first load, the statistics, entry, record and profile screen show error 404. This error does not happen when testing the website locally. Further investigation revealed that GitHub pages do not support single-page-applications. Since Back-end needs to be implemented in future work, this issue can be fixed by finding a web host that supports single-page-applications.

**Improvements**

- The real-time functionality provided by FireBase creates a random key upon POST request, which complicates the handling of data and adds complexity to the system. Firebase offers a storage functionality where schemas can be pre-designed, avoiding generating random keys and simplifying data handling. Research, learning and understanding this Firebase functionality is needed to assess whether this could be a viable solution.
- Tangible must be hosted by another provider when the back-end is introduced. Currently, Git-hub pages do not support back-end hosting.

**Difficulty**

Applying this component required extensive investigation, but the implementation was straightforward.

## 3.4 Must have user Sign-in

Initially, user sign-in was an optional functionality, but it became a Must have to specify user roles, i.e. (spectator, pain user) and fulfilled other must-have functionalities.

**Meets project specification/Functionality**

User sign-in was successfully implemented hence complying with project specifications. Sign-In functionality has been successfully implemented.

**Obstacles**

- Figuring out how to manage user roles and fetch user information was a big challenge that required careful thought and critical thinking. To solve it, when a user signs in, a profile is created with the user information, including chosen role and ID. This profile is fetched on user login, which required a complete understanding of synchronous and asynchronous behaviours.
- All the screens in the application needed to be aware of the user ID and chosen role but passing the information between components proved too complicated and untidy. For this "Context", a React technique that allows the system to access variables throughout the application was implemented.

**Bugs**

- No bugs detected

**Improvements**

- Inputs need validation

**Difficulty**

Due to the time spent learning about "React Context" and dealing with asynchronous and synchronous code, this task proves to be daunting and lengthy. In contrast, the interface for the home page was a template borrowed from Material UI from the source: https://material-ui.com/getting-started/templates/, making the interface part simple to achieve.

## 3.5 Must allow pain users to be spectated by another member. Should have a user guide. Should have alert options. Could have reminders. Could export data. Should have customizable pain scale.

Since all of these objectives belong to the profile screen, they will be reviewed together.

**Meets project specification**

- The should export data has been removed from the application because the spectator already allows the user to share the information and seemed redundant.
- The could have reminders functionality has been removed because there is no need to set a reminder for a pain entry as this is meant to be recorded when a pain user has a pain episode.
- The should have alert functionality has not been implemented due to a shortage of time.
- The should have customizable pain scale functionality has not been yet implemented due to a shortage of time.
- The spectator functionality meets the project specifications because it allows a pain user to add spectator users to spectate their accounts.

**Obstacles**

- No obstacles emerged during implementations

**Bugs**

- No bugs were detected.

**Improvements**

- The spectator functionality can be significantly benefited by adding spectator settings as indicated in this project's mock-up. These settings can allow the pain user to choose what he wants to share and with whom.
- The interface for the spectator functionality has been kept simple due to a shortage of time; this should be enhanced.

- The app currently does not support more than one spectator per pain user; this needs to change to allow multiple spectators, allowing the user to include as many people as they desire.
- Currently, many fetch requests are taking place to perform this functionality; this is due to a random key generated by Firebase. More research is needed to find a workaround and reduce the number of requests sent to the server.

**Difficulty**

This functionality was applied later in the project when most of the methods used were already researched and known to the developer, making the implementation lengthy but straightforward.

**Functionality**

Spectator functionality has been fully implemented.

## 3.6 Must implement the interface agreed in the investigation conceptual diagram.

A reference to the conceptual diagram mentioned in the title can be found in the Appendix C.

**Meets project specification/Functionality**

Most of the functionality proposed in the conceptual diagram has been implemented but does not follow the same design. The application was meant to be usable by smartphone and computer users. The conceptual diagram is based on smartphones, but the computer version was built instead due to the shortage of time to learn phone base technology. Other changes also include:

- Creation of "Home" screen to integrate the sign-in and login functionality.
- The design shows log history on the statistic screen to save space, but the computer screen size allows for more space; hence "Records" screen was created where the user can access log history.

To conclude, the conceptual diagram functionality has been successfully implemented, although not with the same design.

**Obstacles**

- The user guide was based on the mock-up. Since the interface has changed the user guide will need to be updated.

**Bugs**

- No bugs detected

**Improvements**

- The dates should be shown in stack order to introduce the user to the latest entry first. For this, the array which contains the information needs to be iterated backwards.

- The content on the page grows the more entries it holds, making it hard to manage. If date containers could be minimized and make the content scrollable, it would be more user-friendly and easier to navigate.
- Update user guide

**Difficulty**

The records screen needed a high level of logic, data handling, and date format methods, which prove to be a daunting task. The interface was also complex to build, making the process lengthy.

## 3.7 Must include accessibility and usability measures for the elderly.

This functionality has been mistakenly skipped in the project investigation report; hence has been introduced in this report. This section will feature a checklist with the measures researched; if needed, it will explain how the measure was or why it was not implemented.

☒ Font size must be 19 px for a computer screen and 16 px for a phone screen.

- Material UI provides functionality to change the font size of "body" type text for all the application. Queries have not been implemented yet to change the text to 16 px on the phone, and it remains on 19 px.

☒ Font type must be Roboto

- The default font type of Material UI is Roboto.

☒ The contrast ratio must comply with the Web Content Accessibility Guidelines.

- The developer tools in google chrome come with a functionality called "Lighthouse", which analyses the application and tests for contrast ratio, giving Tangible a score of 98/100 see Appendix D with the only problem being the navbar and the text in it. Navbar colour needs to shift to darker to fix this issue.

☒ Paragraph length must be 50-60 characters, and line separation

☒ Typing has been kept to a minimum, and most user input is through selection.

☒ The interaction with the device has been kept short and straightforward; recording a pain episode should not take longer than a minute.

☒ Text will not be overlayed on top of images or graphics unless it is contrasted.

☒ The use of multiple fonts has been avoided.

☒ Proper font scale for consistency.

- Material UI provides its own font scaling

☒ Red and green have been avoided as font colours due to these colours being the hardest to read for colour-blind users.

☒ A friendly tone has been used as a language for the application, which makes it more social.

☒ The border of UI components has a large border radius for a fun, playful feel.

☒The contrast between components will create depth and allow the user to differentiate the interface better.

☒Buttons or other input targeted components must be of the minimum size of 44 by 44 CSS pixels.

☒Headings must use the correct HTML tags; this enhances accessibility helping all users understand the content, and search engines use them to understand a site's content.

☒The feel and look of the app aim to give chronic pain users a healthy, positive, fun, supportive and interactive environment. The logo for the application is a fruit, "tangerine"; this is because both words "Tangerine and Tangible" can be easily associated by sound and letters hence easily remembered. Subsequently, fruits relate to concepts such as healthy and positive, which the app portraits.

☐ The application includes informational tooltips and a user guide for step by step instructions on how to use it.

- Due to time constraints, this functionality has not been implemented.

☒ The pain entries are recorded linearly where the system takes the initiative of asking for the information the user must input.

☒ Scrolling has been avoided in most cases or kept short.

☐ Symbols have been paired with text in most cases.

- The edit and delete options within the record screen does not contain text and could be unclear for some elderly as to what this means. Further analysis is needed to figure out a solution to this issue without clogging the interface.

☐ Actions have been adequately labelled, and feedback, when the action is taken, has been provided.

- Due to time constraints, the system has no feedback when the user submits the records form, deletes, edits, or adds any factors or pain locations.

☒ The app header menu on the top informs the user of its location and the previous location he came from.

## 3.8 Significant changes?

Many changes have been described in the previous sections of chapter 3 of this report, such as creating the record and home screen or adding and removing certain functionalities. Two more changes have taken place, which will be mentioned in this section.

First, the colour of the application has been changed. The application still uses the colour tangerine for some options, but it is no longer the primary colour, as shown in the mock-up and explained in the design section. Fitting this colour on the application required a few changes that would take too much time. Due to time constraints, the current colour pallet (cyan, tangerine) has been applied.

The second change takes place in the project architecture. Initially, the project was going to be implemented using MVC and a back-end, as stated in section 2.1 of this report. Due to time constraints, there was not enough time to study the back-end technologies, and the application's business logic has been moved to the front-end. The application works well, but many security issues come with exposing the business logic and APIs that need to be addressed. As a result, the project application has taken the form in Figure[figure]. Under the new system, the first step takes place when the user access the app for the first time, step 2 and 3 happen when a fetch request is made to the Firebase server, and this information returns to the app.

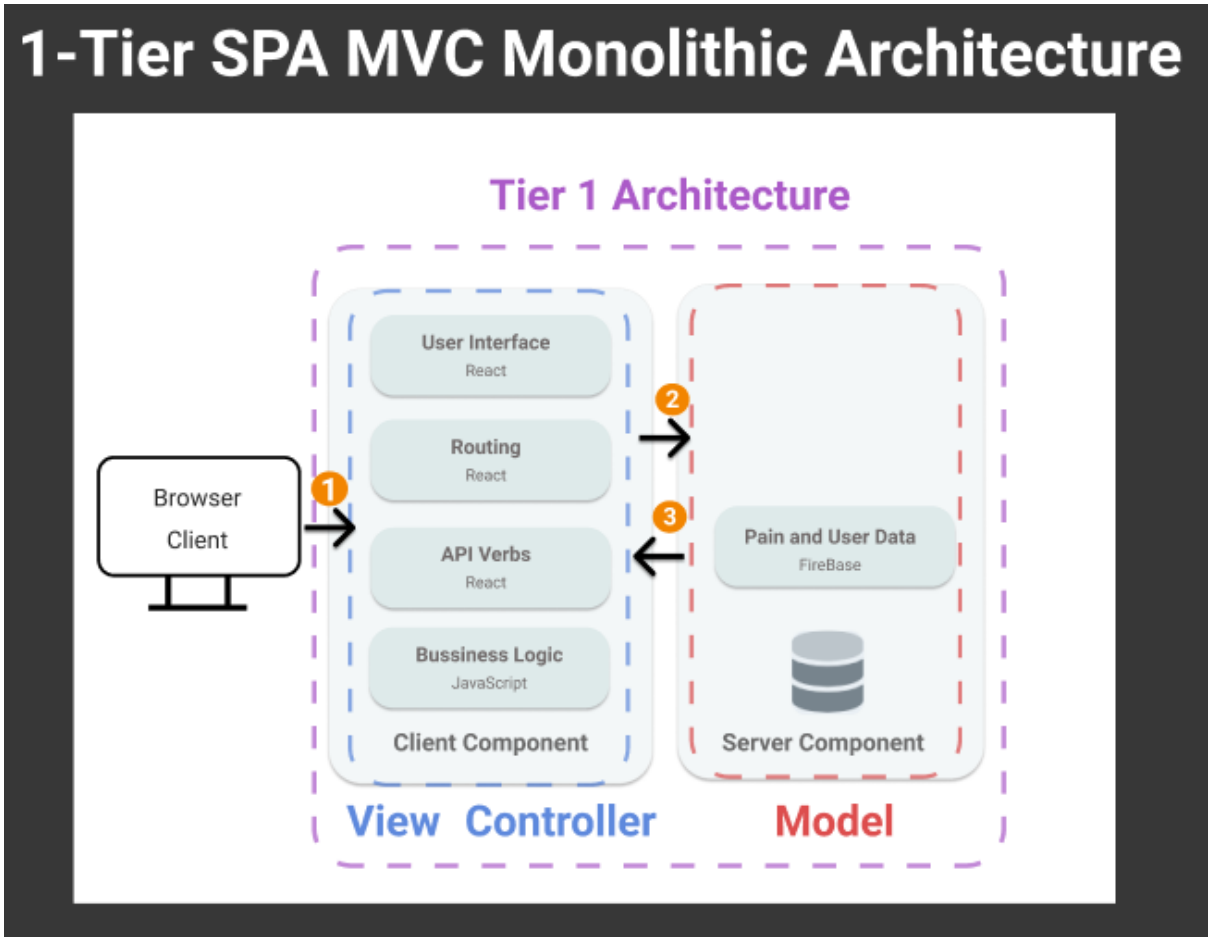



*Figure 8: 1-Tier SPA MVC Monolithic Architecture alternative version*

## 3.7 Testing for Interface, Functionality and Security

Due to time constraints, only certain testing phases were applied to Tangible. This section aims to explain how these testing phases were implemented.

### 3.7.1 Functionality Testing

This type of testing happens in the source code. It is based on the functionality of the software. Each functionality is built by one or more functions. Each function and line within a function has been tested with “console.log”, which outputs the current state or value. Functional testing has been applied by applying this method across all functions and ensuring each response had the desired outcome.

### 3.7.2 Interface Testing

This type of testing happens in the interface of the application. All interactive buttons and interface changes have been tested by interacting with them through the interface and work as intended. Unfortunately, there is still no validation or HTTPS error handling on the interface and checking whether errors are well displayed in the interface is part of this testing which will need to be applied in later updates.

### 3.7.3 Security Testing

This step makes sure that the application is protected against unauthorized access. These URLs, which require user login and unique user ID, have been tested by entering the URLs into the search engine. Unless the user has created an account and is currently logged in, these pages will not show; instead, they are redirected to the home screen for account creation.

# Chapter 4

The previous chapter described and evaluated the project achievements and implementation. This chapter will reflect on the project management techniques. Lastly, the section will highlight LESP issues related to the project.

## 4.1 Reflection of Project Management

The project management applied in this project has been explained in section 1.5 of this report. In the Appendix A I have the documentation of every sprint on how the project was managed. This documentation can help understand what made the project better or worse and make the right adjustment for the next project. Some of the lessons learned are written below.

The list of requirements that needed to be presented determined what the application needed to have and clarified the project's end goal, making waterfall the best project life cycle. Using Trello to break down these milestones and assign dates to them, I determined a start goal and paced myself to work on all project parts effectively regardless of whether that section had been finished. Without this structure, I could have overspent more time on specific project parts, leaving others unworked.

Due to the objectives, start and end goal being clear, I felt updating the project management was not worth the time; this could be perhaps because the project was small or because the direction was already stated and there were not many changes. I benefited myself much more by using it as a guide. I believe that it will require more practice on my behalf to learn how to be more effective with the project management skills and tools used. I can see how a few iterations can develop good habits and a better practice hence greatly benefiting from project management techniques.

## 4.2 Tangible vs Other Pain Management Software

During the investigation report in section 4.2, five pain management apps were reviewed based on a particular criterion. Part of this criteria required similar software to be top rated by other users in the UK google store. Two of the apps mentioned (Bearable and Manage my pain) have great functionality and a stylish user interface. These apps also offer insightful statistics and multiple user settings. The other three apps in line have poor variable recordings, statistics, or no statistics at all. The gap between these apps and the top 2 is pretty big. Based on functionality and aesthetics, it is my personal judgment to say that Tangible is the 3rd best app in the store. This is considering that Tangible still needs a week's work and a version for the phone.

## 4.3 Start over again

If I had to start this project all over again with the current level of expertise, I would skip doing the mock-up interactive. The flow chart was of no help; it seems as if it would be of use if the project needed to be presented or explained to other persons. In exchange, I would spend that time to complete the React component tree as this chart enables me to understand how the application is built and how the information moves between components allowing me to improve the system and reduce the amount of code by reusing redundant components. Since I know a lot about React, Material UI and other third-party components used for the app, I would like to use this time to learn the storage feature of Firebase to void the issues with randomly generated keys from their real-time database services. I would also like to learn Redux and apply it to the project to reduce the amount of code and extra components needed to organize and transport the data between components. Lastly, the application architecture also added no value to the project; it seems worthwhile when the application has to be explained or understood by other programmers; on the other hand, having this information makes it clear what type of application is being built hence is easier to find help on the internet although in this case was not used.

## 4.4 LESP

### 4.4.1 Social and Legal

This section will address possible legal issues found in Tangible.

**Third-party software licensing**: There are no copyright issues with tangible because the third-party software used is open source and allows users to use the content commercially. The following link contains the terms of use of all the software downloaded through their service (NPM), and no issues have been detected.

NPM Packages such as (React, Material UI, Recharts, Calendar and time picker): (GitHub Inc., n.d.).

**Trademarks laws:** The website for the free logo used in Tangible enforces that for commercial use in websites, the developer must add on the footer of the webpage the following line "Icon made by freepik from (Freepik Company S. L., n.d.). This needs to be added to tangible once under commercial use.

**Data Protection and Privacy**: For data protection in the EU and UK, we apply the GDRP or General Data Protection Regulation, a law that governs how data is processed. After Brexit, there are now two regulations, one for the UK and one for the EU. There are some differences between regulation, but the main rules that applied to all data handling are as follow:

- ✓ Inform user how he's data is handled
- ✓ Can't charge a fee to handle right of access
- ✓ If user decided to delete their data, it must be erased
- ✓ User can rectify incorrect data
- ✓ can restrict you from processing their data
- ✓ Users can move their data at their leisure
- ✓ User can object to how you use their data and ask you to stop
- ✓ Data must be collected only for a specific purpose and stored as long as necessary
- ✓ Data security, integrity and confidentiality must be ensured appropriately.

These are some of the main rules that will need to be enforced within Tangible once the application begins usability testing or is commercially deployed. Further research needs to occur to enlist all the possible data protection issues that need to be enforced by tangible. The application also needs to demonstrate compliance with the GDPR.

### 4.4.2 Social and Ethical

Taking care of ethical matters help protect the organization and developers from legal actions, gain goodwill within the community, and avoid defamation.

Since Tangible is a personal project, not many ethical issues apply, as would be the case if it was developed by a team and sponsored by a client. However, software piracy does apply to Tangible, although as stated in the Legal section of this chapter, all third-party software used infringed no copyright laws. Tangible also aims to apply GDRP, meaning all data will be handled ethically. Lastly, at the start of this project, and ethics form was filled stating many ethical facts that have not changed, but once the application reaches usability testing and commercial use, this form will have to be updated.

### 4.4.3 Social and Professional

Most professions have a professional body relating to the responsibilities and trust exert on the individual. For software developers, the British Computer Society has a code of conduct, "The code of conduct for BCS members". These rules set the professional standard to direct the behaviour of its members. The code of conduct can be found in the following source: (BCS, The Chartered Institute for IT, 2021). Adherence to the code promotes professionalism, public interest, loyalty to the employer and professional field, and the individual's level of competence.

## 4.5 Conclusions and future work

This project helped me to identify project management tools and the effectiveness of their implementation. I have learned that more experience is needed and perhaps the use of different project management tools to build a habit that will benefit me to manage my projects better. Many different techniques were applied during the design process of this project. I learn that one of the most helpful design concepts to have is a mock-up as it gives visual direction to the implementation section. I also learned not to create a User Guide from a mock-up, which can change drastically during implementation. In the beginning, I was trying to make the application look exactly like the mock-up but learned to embrace change. During the implementation part, I found building a react component tree helpful, but due to the shortage of time, I could not complete it; this tree will need to be finished to identify improvements for future work. The project also taught me how to remain disciplined and work on the important parts of the project and dedicate enough time to the different parts of the project life cycle. Lastly, I learned about LESP issues and the importance of keeping the organization and the individual out of legal problems while showing ethical and professional skills that benefit the individual and the organization, co-workers, and users.

Tangible has not been completed yet; a week of work is expected in order to finish the following functionalities and fix bugs:

- Spectator settings
- Customizable pain scale
- Include more statistical components

There is also a list of improvements that can make Tangible a more robust software and better experience for users; these improvements are as follows:

- Apply Material UI Grids to all the app, so that it is fully responsive at any screen size.
- Create one reusable component for treatment, medicine and location.
- Minimize the number of GET requests by loading all needed information on user login.

- Add Redux to the application to avoid passing props between parent and child components.
- Find a workaround for the Firebase random generates keys to reduce the number of fetch requests for the key value.
- Add a backend server that handles the business logic and API calls to make the app more secure.
- Perform Usability testing.

In conclusion, the project achieves its most important objectives. Tangible contains the top features of other pain management apps and offers a novel feature, "spectator". The next step is to test its usability with real users to determine whether this new feature can make the application stand out. A positive result means that the application has a future hence investing more resources for the development of its commercial is necessary to allow for a solution that manages their user's pain and helps them and their significant others to be on the same page.

# Bibliography


BUSTAMANTE MIRAYO, A., 2012. *Pain diary : Pain management platform*.

Agile Manifesto for Software Development. (2015, June 29). Retrieved July 8, 2021, from Agilealliance.org website: https://www.agilealliance.org/agile101/the-agile-manifesto/

BCS, The Chartered Institute for IT. (2021, July 6). BCS, THE CHARTERED INSTITUTE FOR IT CODE OF CONDUCT FOR BCS MEMBERS. Retrieved August 24, 2021, from Bcs.org website: https://www.bcs.org/media/2211/bcs-code-of-conduct.pdf

Borough Clinical Commissioning Group. (2018). Managing Patient Expectation. In Borough Clinical Commissioning Group (Ed.), GMMMG Opioid Prescribing for Chronic Pain: Resource Pack. Retrieved from http://gmmmg.nhs.uk/docs/guidance/Final-Opioid-Resource-Pack-Approved-CSB-August-2018.pdf

Freoksenet, Anoobis, & Wojtekmaj. (2021). React-calendar. Retrieved from https://www.npmjs.com/package/react-calendar

Freepik Company S. L. (n.d.). Free Vector Icons and Stickers - Thousands of resources to download. Retrieved August 24, 2021, from Flaticon.com website: https://www.flaticon.com/

GitHub Inc. (n.d.). npm Open-Source Terms. Retrieved August 24, 2021, from Npmjs.com website: https://docs.npmjs.com/policies/open-source-terms

Google. (n.d.). Material-UI: A popular React UI framework. Retrieved from https://material-ui.com/

How to design a web application: Software architecture 101. (n.d.). Retrieved July 8, 2021, from Educative.io website: https://www.educative.io/blog/how-to-design-a-web-application-software-architecture-101

NodeJS - The Complete Guide (MVC, REST APIs, GraphQL, Deno). (n.d.). Retrieved July 11, 2021, from Udemy.com website: https://www.udemy.com/course/nodejs-the-complete-guide/

React 16: The Complete Course (incl. React Router 4 & Redux). (n.d.). Retrieved July 11, 2021, from Udemy.com website: https://www.udemy.com/course/react-the-complete-guide-incl-redux/

Recharts Group. (n.d.). Rechart API. Retrieved from https://recharts.org/en-US/api

# Appendices

## Appendix A

This appendix includes the weekly reports / follow up of the project management plan.

### Week 1 between 28 – 2 July

This week was used to take a crash course on React.

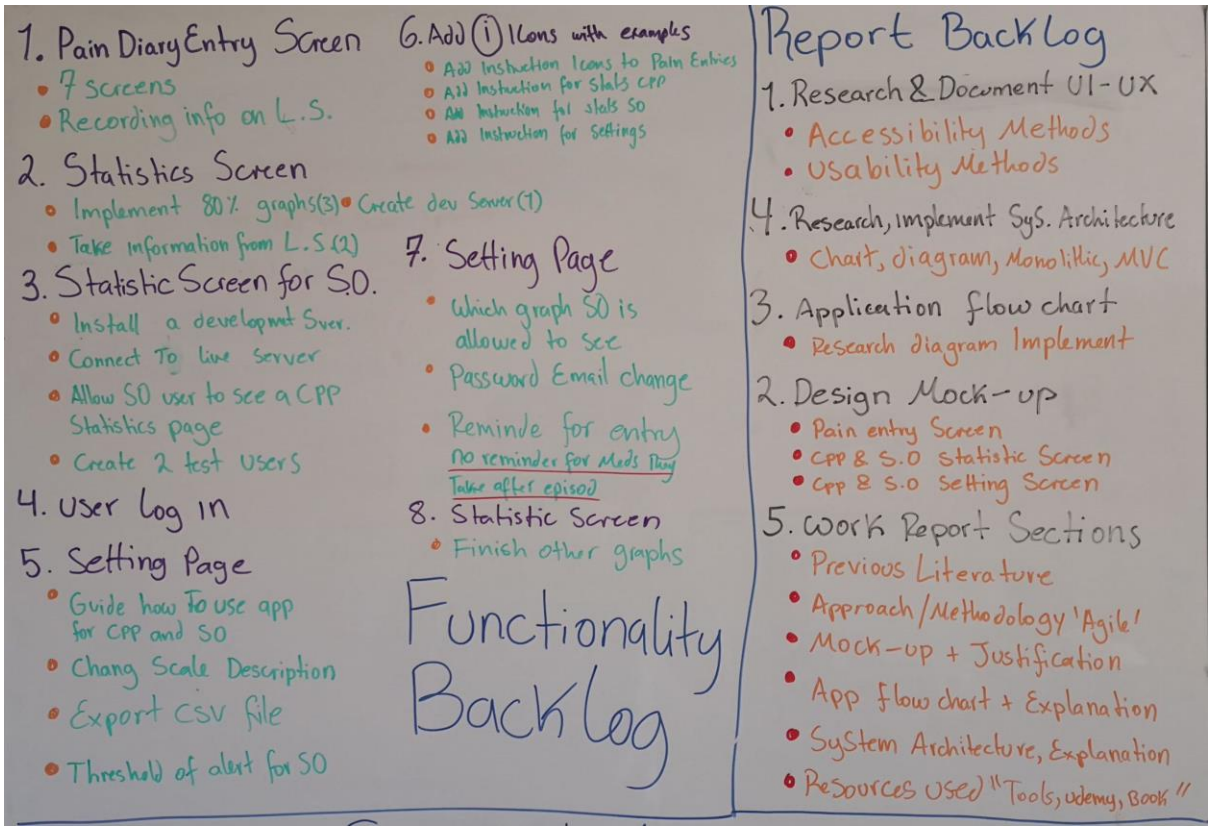
## Sprint Two

## Week 2 between 5 – 9 July

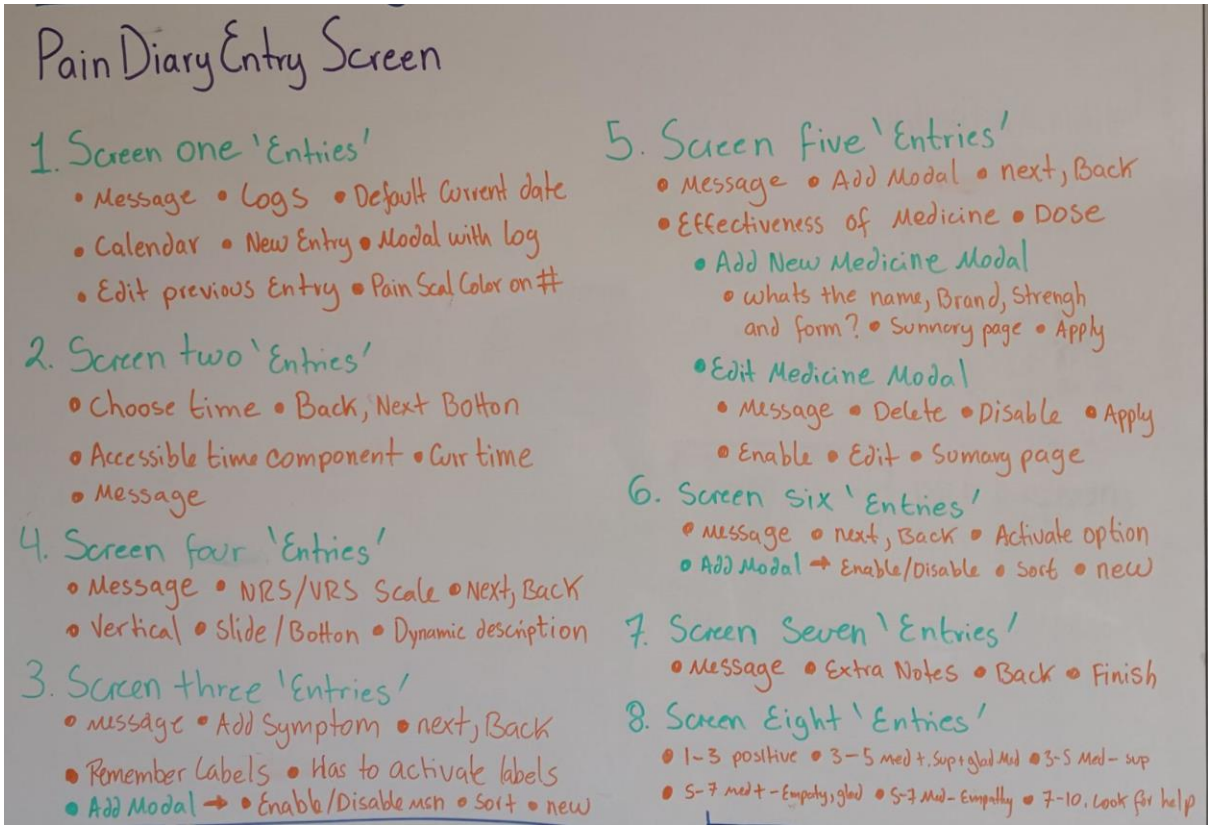
This week was used to implement project management strategies, plan for the project and work on the design aspect of it.

Tasks and functionality for the implementation of the app, report and project management where analysed and written down on a whiteboard in order to incorporate them to the project management plan.

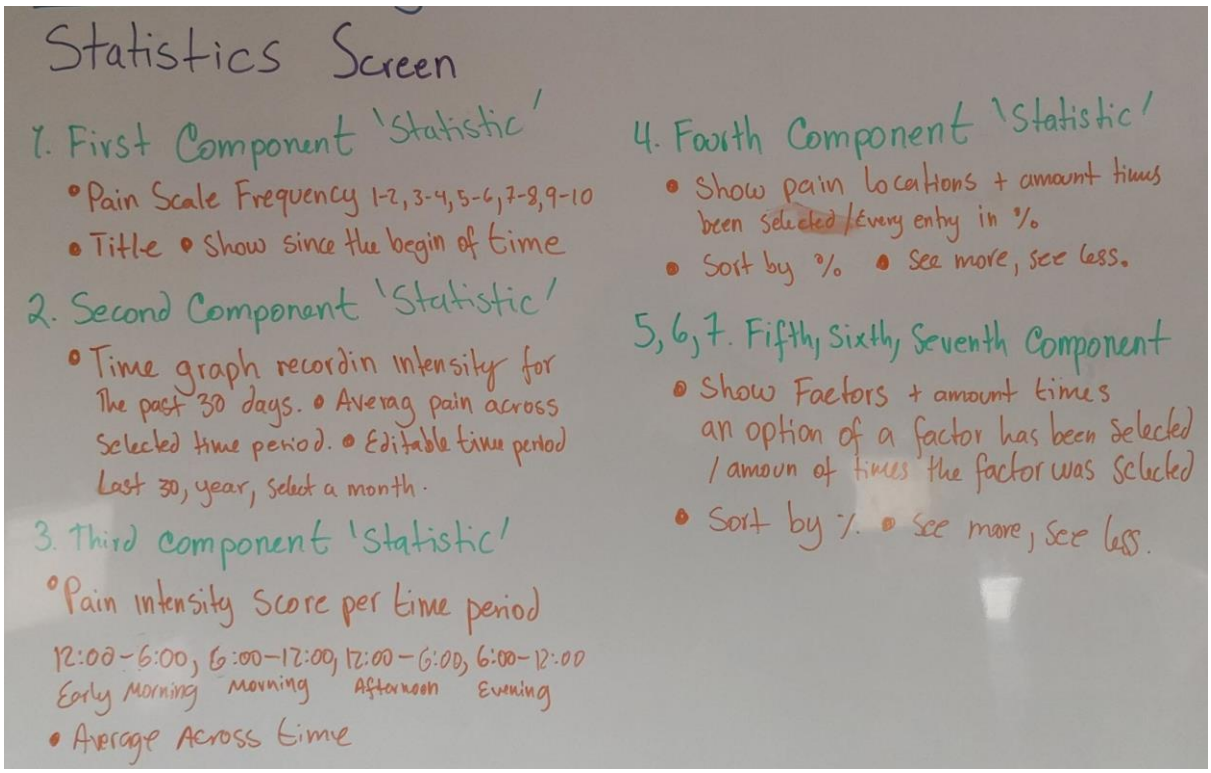
### Breakdown of project plan



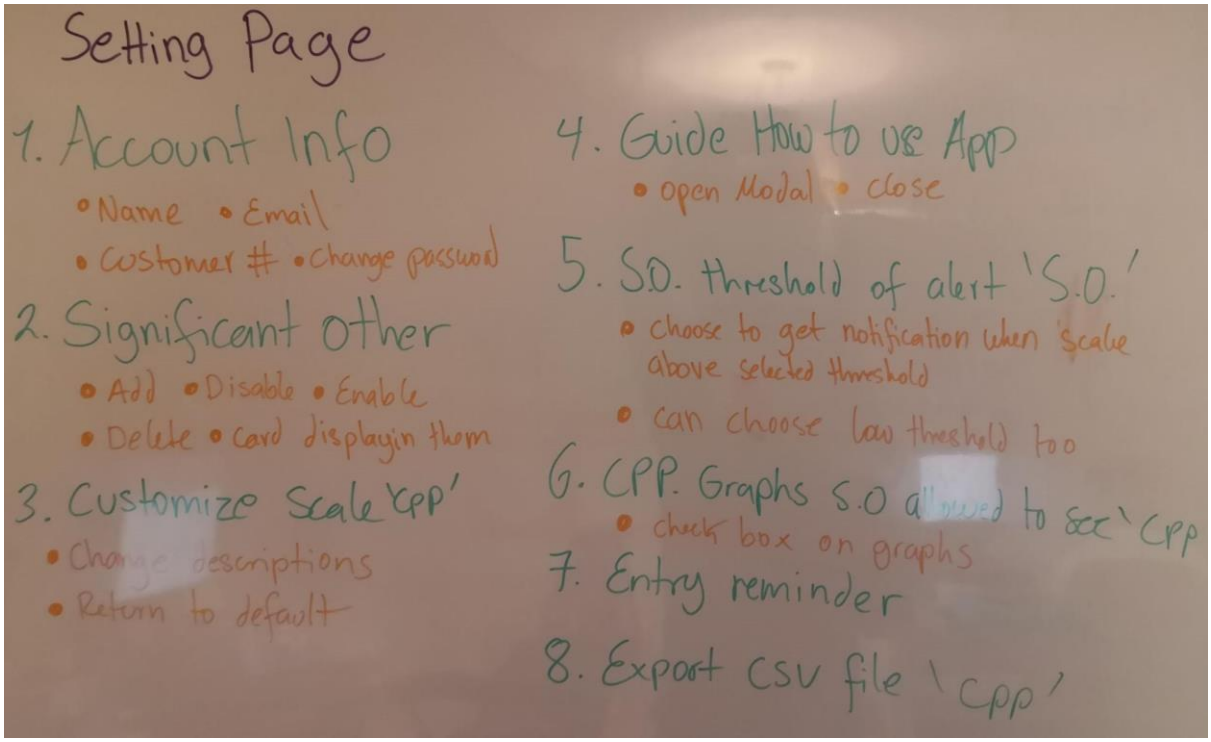

### Entries Screen Mock-up plan



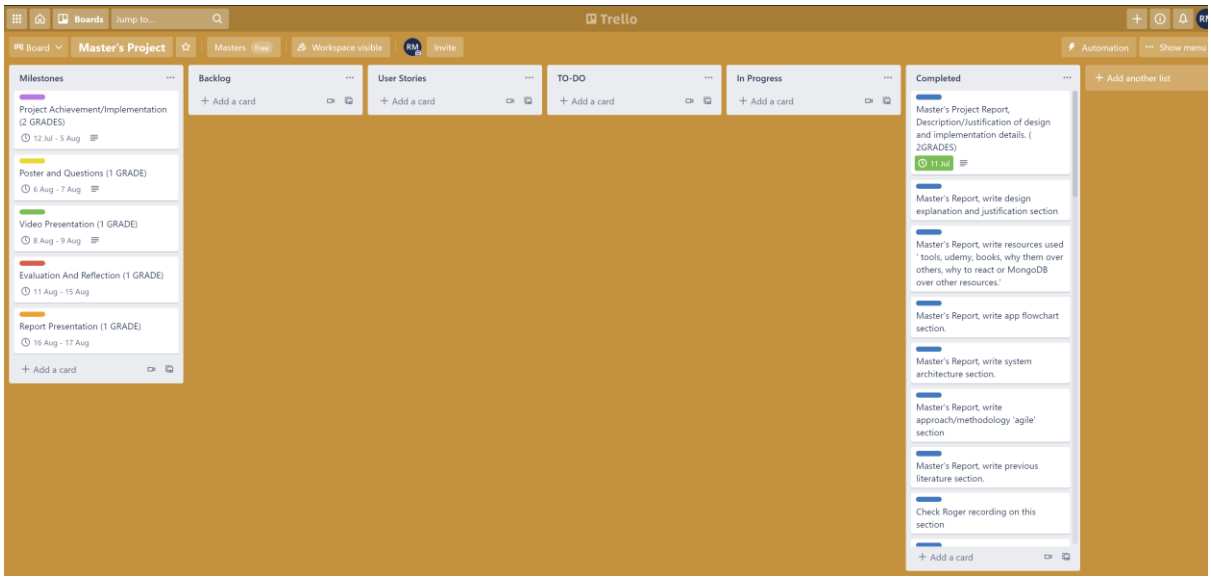

**Statistics Screen Mock-up plan**




**Settings Page Mock-up plan**




## Week 2 Trello after Desgin phase




## The following tasks were completed during this week

- Research & Document UI-UX Accessibility and Usability Methods
- Create Functionality goals
- Create report structure for design chapter
- Create Mock-up structure
- Create project management on Trello
- Build user interface mock-up in Figma

- Build an application flowchart in Figma
- Design System architecture in Figma
- Started writing the second chapter of the report
- Test if all interface components work properly on the app
- Test if the business logic and front-end logic is working properly on the source code

### Obstacles

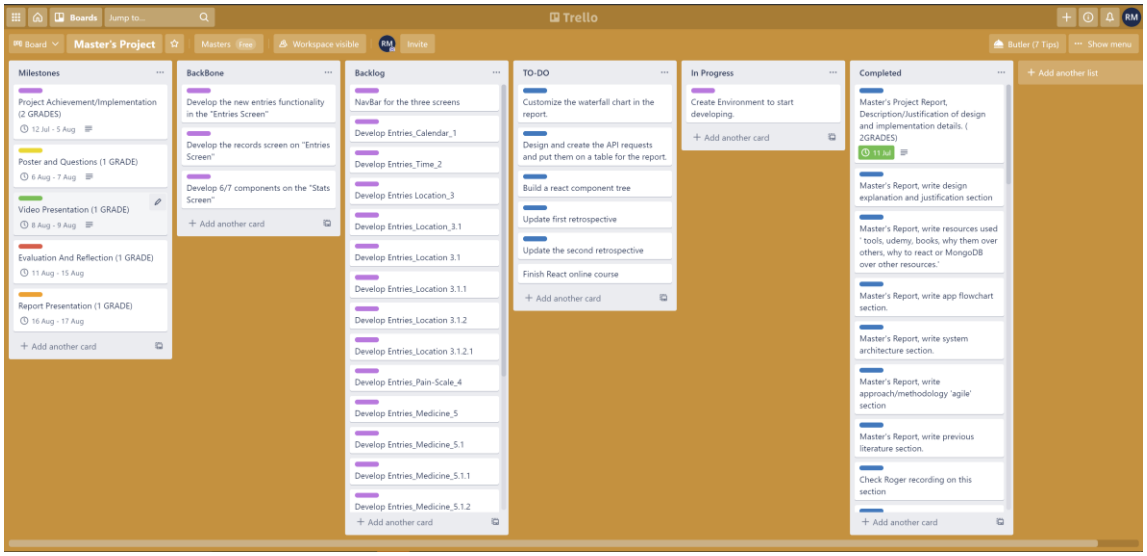
Agile as app development cycle did not fit the current flow. This problem was fixed by adopting a waterfall software development life cycle.

Current knowledge of react is not enough to complete the application, react course needs to be completed.

### Plan for next week

The plan for next week is to start building the application and complete the React course and finish writing chapter 2 for the report. The backlog has been updated with the backlog created on the previous week in the white board. User stories have been created for more than a week's work. Extra sections for chapter two have been identified and added.

## Week 3 Trello planning for Implementation



## Sprint Three

## Week 3 between 12 – 16 July

This week was used to start the coding environment for the project implementation. Also finished writing the second chapter of the report. During this week the react crash course was **completed.**

**The following tasks were completed during this week**

- React online course was completed
- Finish Chapter 2 of the report
- Started the coding environment for the application
- Navbar for the application was completed
- Test if all interface components work properly on the app
- Test if the business logic and front-end logic is working properly on the source code

**Obstacles**

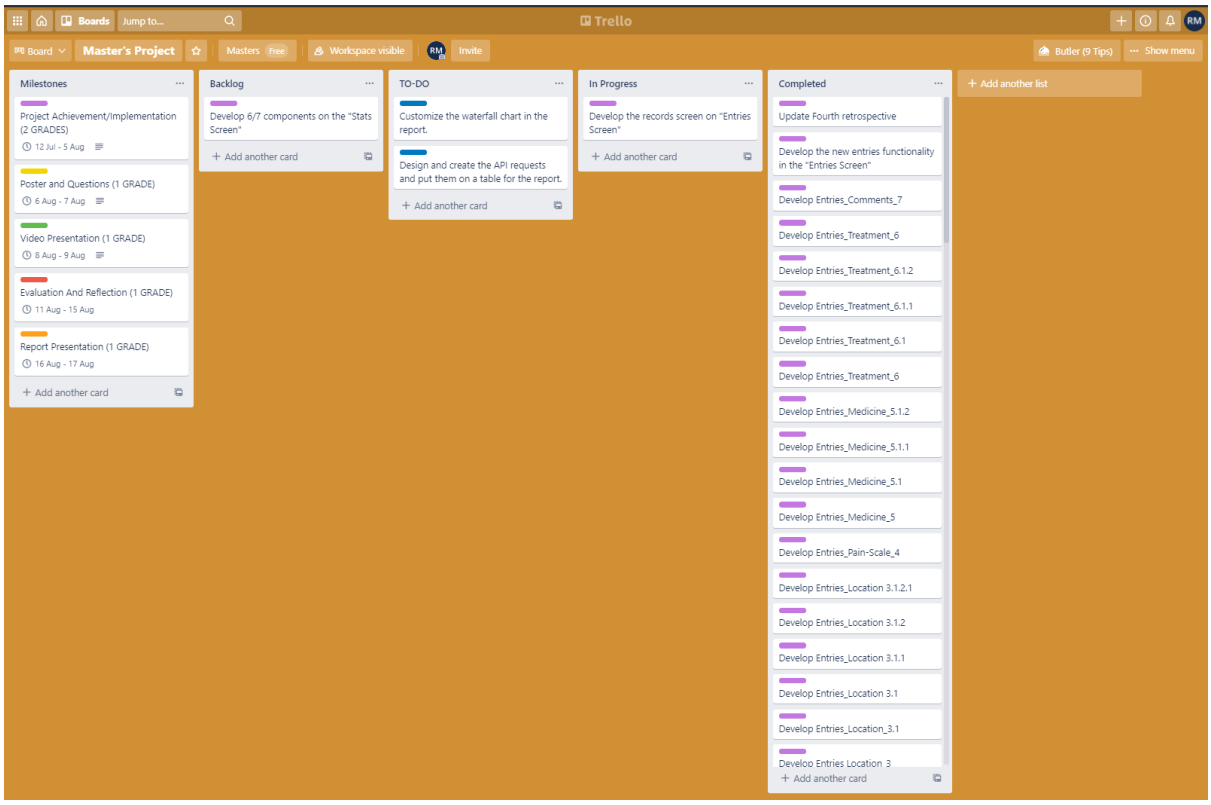
- Creating the application using just CSS is too time consuming, a framework needs to be implemented.
- Not enough React knowledge to build the application. Must study more of the online react course.

# Sprint Four

## Week 4 between 19 – 23 July

This week the developing part of the application has been continued. The first screen of the application which gathers user data has been completed.

### Week 4 Trello during implementation phase




### The following tasks were completed during this week

- Pain Entry screen date and time picker completed
- FireBase database has been implemented for the project
- Pain Entry screen pain location completed
- Pain Entry screen pain scale completed
- Pain Entry screen medication completed
- Pain Entry screen treatment completed
- Pain Entry screen extra comments completed
- React tree component has been created
- Test if all interface components work properly on the app
- Test if the business logic and front-end logic is working properly on the source code

### Obstacles

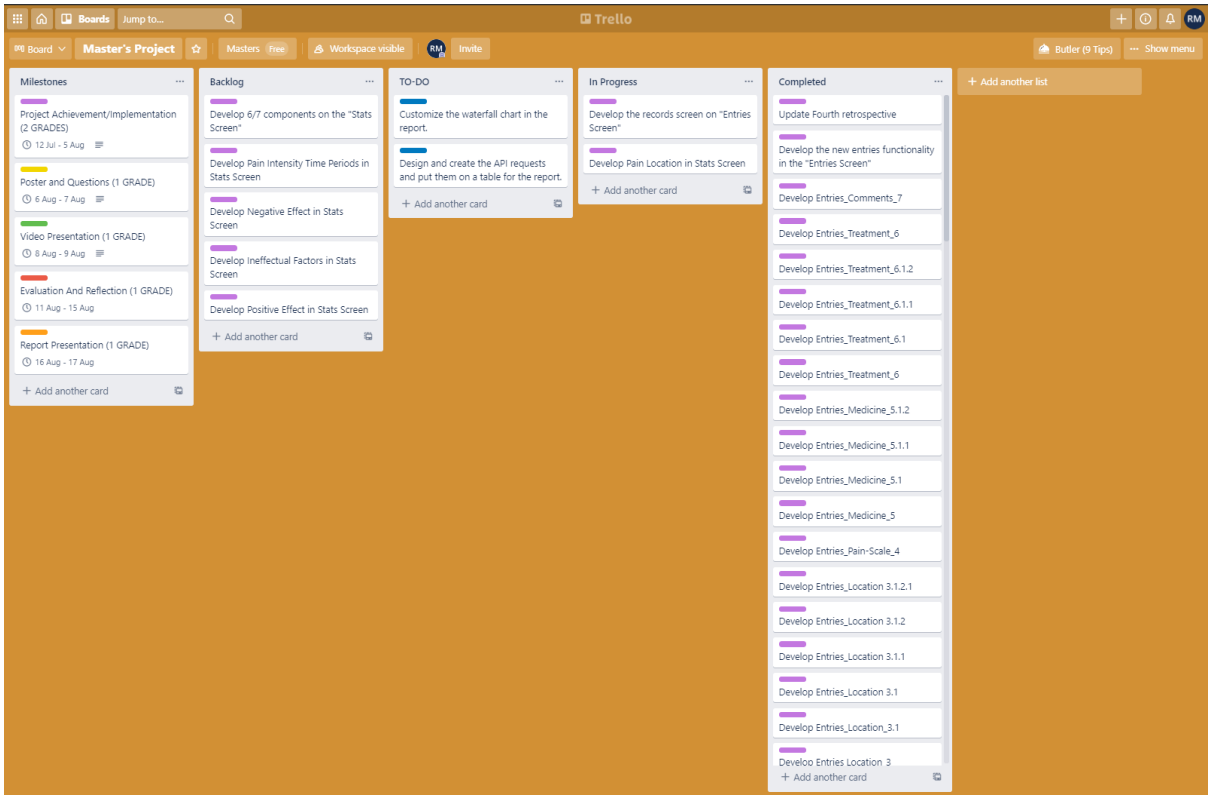
- CSS frameworks for react are very complex, they must be studied thoroughly to be able to implement them properly. There is not enough time to learn a react CSS framework.

With poor framework knowledge CSS components can be used for obtaining functionality but not a good look. Inline CSS will be used to make up for this flaw and allow the components to seem presentable.

- Unable to store values for the multiple pain entry screens. Firebase database was created and used to store the values.
- Unable to implement Material UI calendar for date picker as this won't appear extended and instead minimized. Another calendar with this quality has been searched for and implemented.
- Location functionality has been implemented but it is not clear how to define which components or how information will flow between them. A react component tree has been created to be able to visualize this process.
- Many warnings and errors have appeared in the development tools of the browser. Most of these errors and warnings have been fixed.
- Unable to hide or show a react component when they are created dynamically in Medication and Treatment screens. Research was done and pure Javascript and DOM manipulation has been used to change the display attribute to inline or none in order to show or hide the component.
- Not enough time to implement statistic screen, setting screen, accounts and account types and back-end. Functionality is already being prioritize over looks but core functionality will have to be prioritize over functionality.

**Plan for next week**

Next week the show record functionality and statistics of recorded entries will be worked on. Consideration of back-end in order to include accounts creation and link accounts between pain user and spectator.

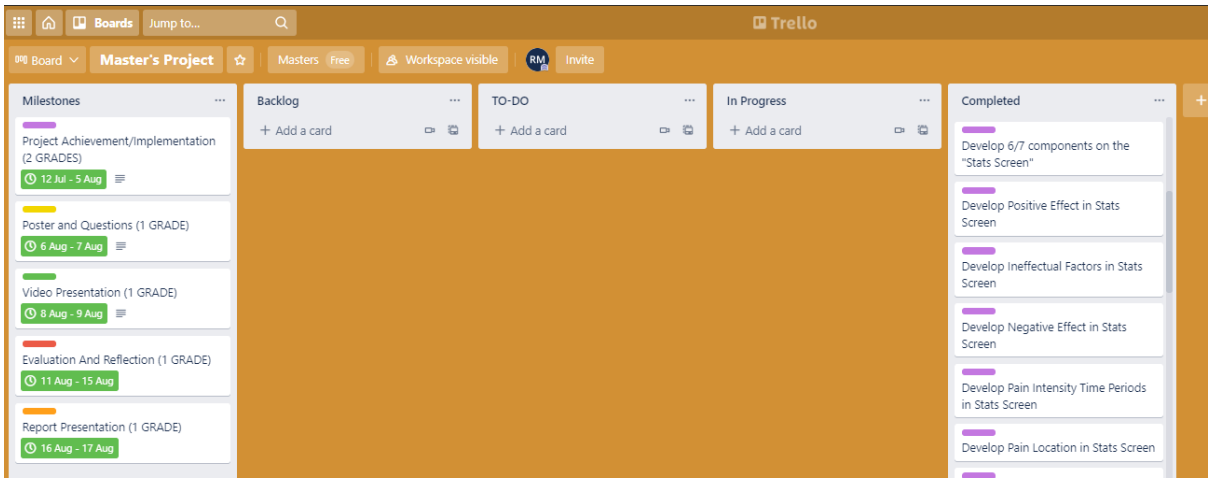


# Sprint 5

## Week 5 between 26 - 30 July

During this week the statistic screen was implemented.

### Week 5 Trello during implementation phase




**The following tasks were implemented**

- Develop pain location frequency chart
- Develop time periods chart
- Develop Negative effect graph
- Develop Positive effect graph
- Develop ineffectual factor graph
- Read Material UI documentation thoroughly and understood more of it's implementation methods.
- Test if all interface components work properly on the app
- Test if the business logic and front-end logic is working properly on the source code

**Obstacles**

- Applying the logic for the algorithms in the graphs was daunting and complex
- Thorough read and understanding of the documentation in Rechart was needed to properly implement the graphs.
- Visualizing that there is not enough time to learn back-end technology for the implementation of accounts, I need a workaround.

**Plan for next week**

- Find a workaround for building account functionality
- Work on the profile screen functionality
- Work on Records screen functionality
- Fix UI spacing between components in all screens
- Update Navbar, with proper Material UI components

# Sprint 6

## Week 6 between 2 - 6 August

Since there is not enough time to learn back-end technologies to apply to the project the business logic has been placed in the front-end. Workaround for account creation has been found, Firebase API provides functionality for account creation. User Interface has been improved aesthetically. New screens have been initialized and Navbar improved.

**The following tasks were implemented**

- Found workaround for building account functionality with Firebase API
- Started the profile screen by building the interface and routing system
- Finished the "User Guide" section of the profile screen.
- Started the record screen page by building some of the interface and routing system.
- User interface of tangible has been aesthetically improved
- Navbar has been updated with light colours, a logo and Material UI components.
- Routing system code logic has to be updated with the new Material UI Navbar
- Cancel button has been added to edit and add options in the entries screen for the location, medication, and treatment section.
- Home screen has been initialized for account creation functionality
- Test if all interface components work properly on the app
- Test if the business logic and front-end logic is working properly on the source code

**Obstacles**

- Makes no sense to add account creation functionality to profile, Home screen has been introduced.
- "User Guide" is based on project mock-up, there will be no time to update the guide to the new system, it will have to be explained in project report.
- Record screen needs a high level of data manipulation and algorithm logic that needs to be carefully thought. Will pass this work for next week.
- When an account is created the new user information is sent to firebase for the first time. When the user tries to login the account information is required, but when the information is fetched from the server is only accessible within the .then() method of the fetch request and cannot be passed into a variable created outside the .then() for further use into the application, work around needs to be implemented.
- Passing the account information across all the components to retrieve specific user information is a lengthy and untidy task, workaround is needed.

**Plan for next week**

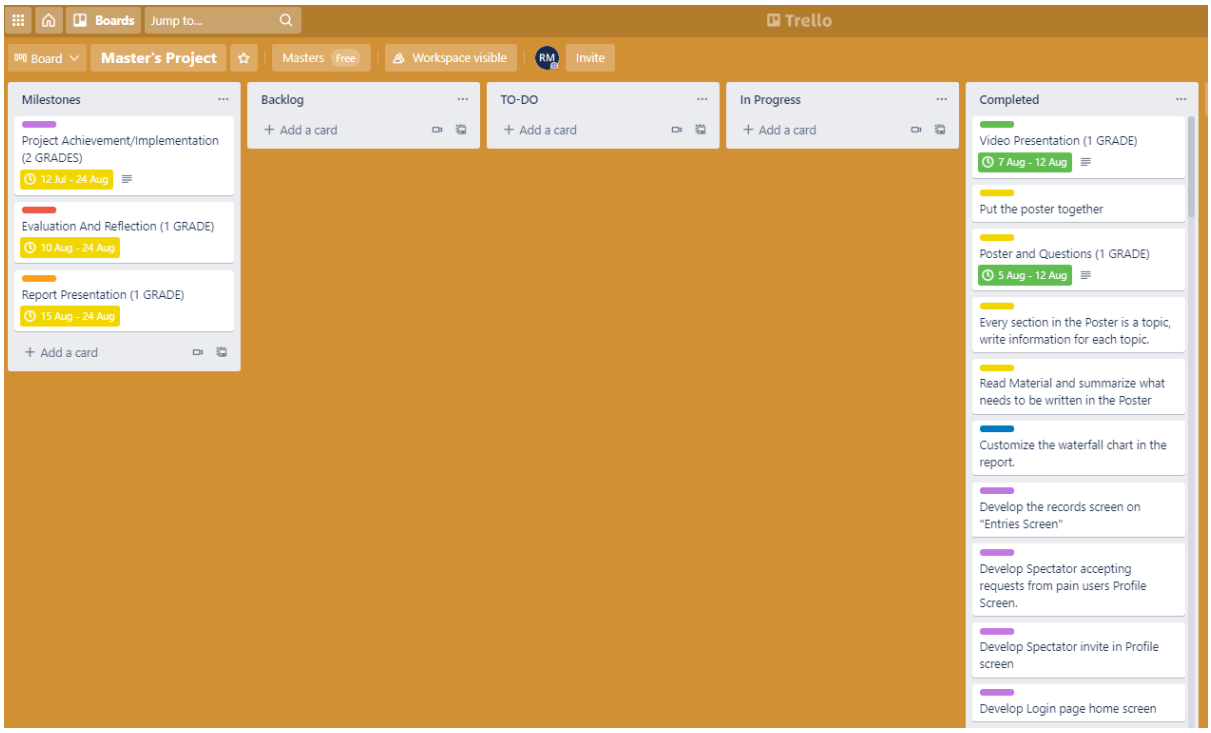
- Find a workaround for accessing the account information on user login.
- Find a workaround for making the account information accessible across all the application.
- Finish working on the interface and business logic of the records screen.
- Finish working on the interface and business logic of the home screen.
- Finish working on the interface and business logic of the profile screen.
- Work on the project poster, presentation, and demo.

# Sprint 7

## Week 7 between 9 - 13 August

During this week the computer suffered a problem that had to do with the BIOS, operating system and programs have to be reinstalled. Coursework extension has been sent to the university. While solving the problem Poster and project Demo and presentation were mostly finished. Workaround has been found for account creation and Home screen has been finished. Not enough time to finish profile screen so only core functionality has been done. Unable to contact supervisor, many assumptions have been made in regards to deliverables.

### Week 7 Trello after Implementation phase




**The following Tasks were implemented**

- React has a concept called context which allows the information to be accessed across the entire application and has been implemented in the app.
- Values can be assigned by using context methods inside then() statements hence fixing the problem from previous week.
- Home screen has been finished quickly by using a Material UI template. User can now create pain user and spectator account.
- Core functionality of profile screen has been finished, pain user can now add and remove a spectator and spectator can remove a pain user and view pain user data.
- Waterfall chart has been customized
- Changes on the project architecture has been updated and new chart has been created.
- Record screen has been finished
- Poster has been finished and delivered

- Presentation and Demo have been finished and delivered.
- Test if all interface components work properly on the app
- Test if the business logic and front-end logic is working properly on the source code

**Obstacles**

- Technical problems with the computer used to develop the application. Second computer has been used to work on the project while main computer was fixed.
- Unable to contact supervisor for advice with deliverables. Work has been presented without clarifications.
- Not enough time to breakdown the next topics of the project in Trello, I don't find more use for the project management tool at this state and with this much time anymore.
- I have to travel to Poland for personal matters, packing and travel planning will take time off the project.
- Second shot of the COVID Vaccine will take place this week, extremely sore arm and lack of energy to be expected.

**Plan for next week**

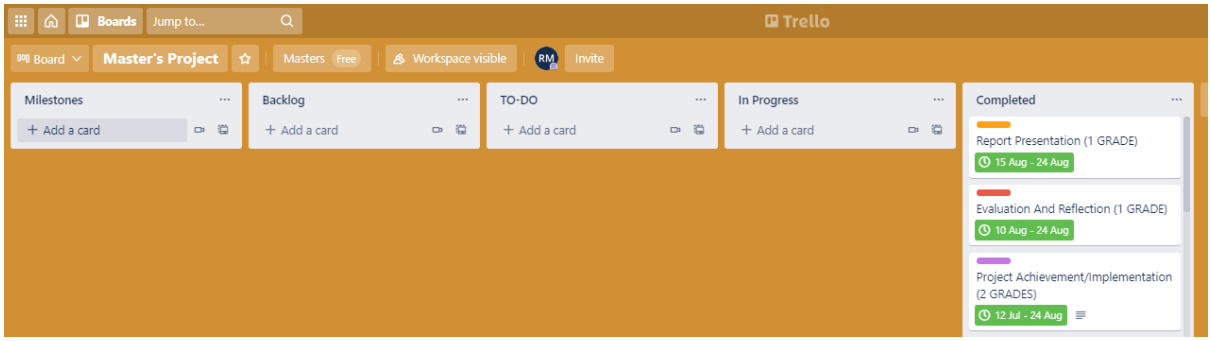
- Update the design section of the report which need to be updated.
- Work on the project achievement and implementation section of the report.
- Work on the Evaluation and reflection section of the report
- Make sure the project report is properly formatted.

# Sprint 8

## Week 8 between 16 – 20 and 20 - 24 August

The extension has been granted by RGU and delivery extended until 24 of August. Not enough time to work on the report as travel planning and had a strong fever of 37.7 the day of COVID vaccine. More time has been lost while traveling and settling in Poland. RGU wont accept a second extension. Nonetheless I have been able to find a space and time to finish writing the project report but quality is not as good as it could have been.

### Week 8 Trello after report




**The following tasks were implemented**

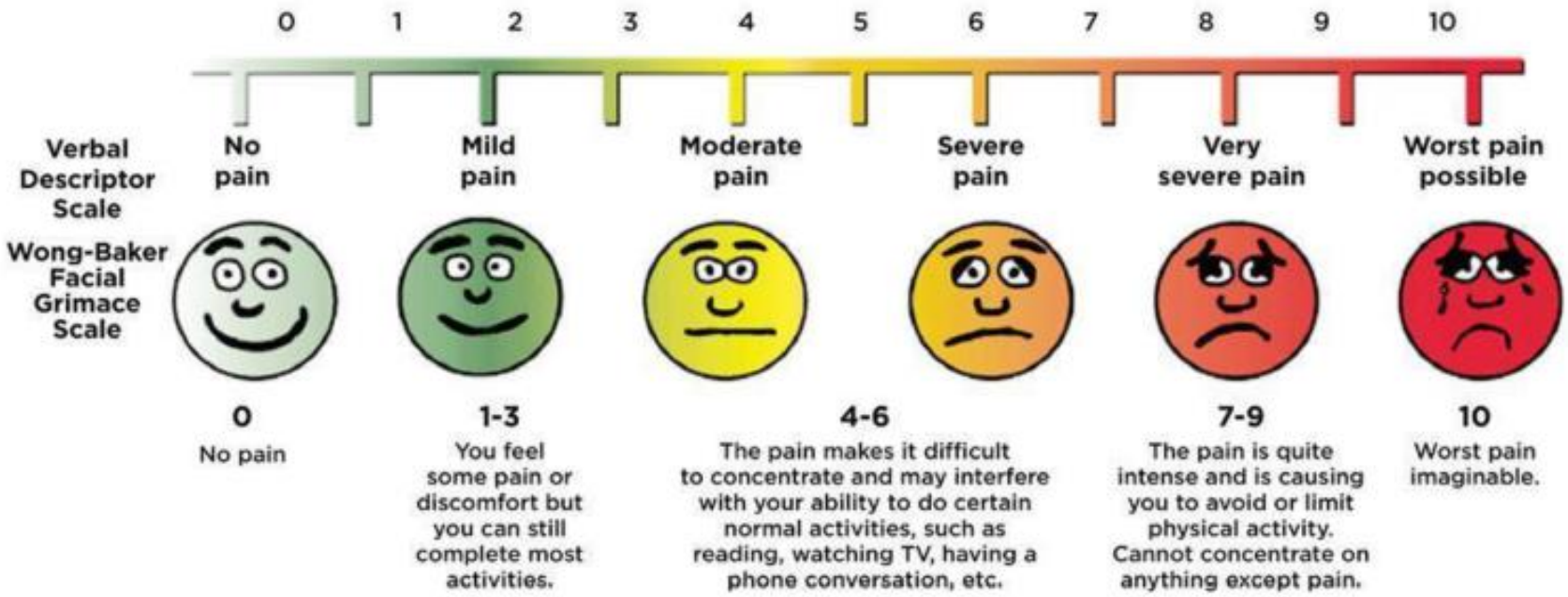
- The project achievement and implementation section of the report has been finished.
- The Evaluation and reflection section of the report has been finished.
- Report has been properly formatted.

**Obstacles**

- Travel planning and packing has taken off time and energy
- COVID vaccine Fever and sore arm has taken time and energy.
- Traveling and settling in Poland has taken time and energy.
- Arrangements were made in time to have a place to work on the report on arrival allowing enough time and comfort to finish the project's report.

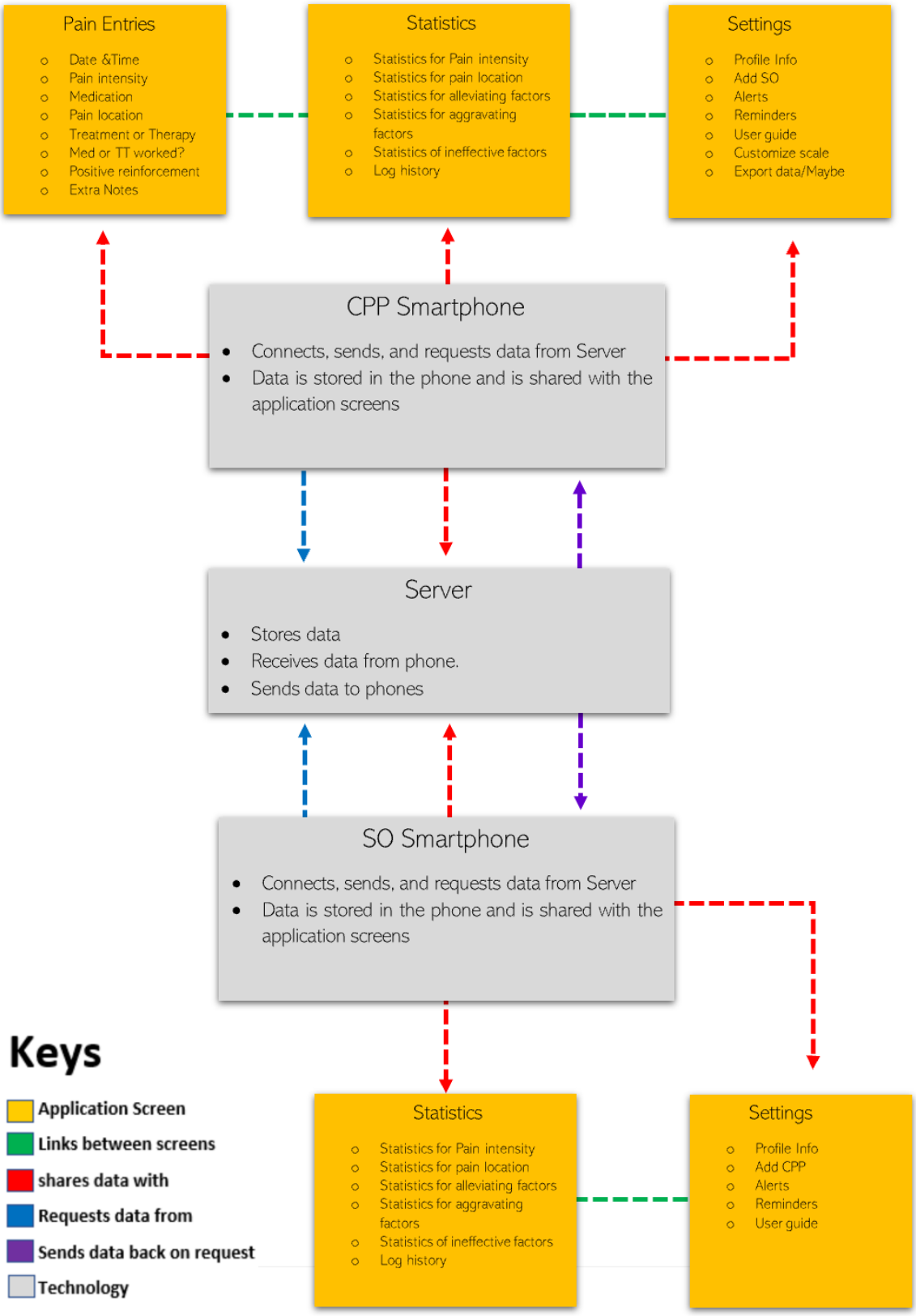
## Appendix B

This appendix shows a hybrid NRS and VRS rating scale by the NHS. (Borough Clinical Commissioning Group, 2018)



## Appendix C

This is the conceptual diagram design during the project investigation phase.



# Appendix D

This is the result from the lighthouse test provided by the developer tools in google chrome. The full test will be provided in the dropbox for extra materials.