COSC345 Software Engineering Project

Harry Ellerm[1], Bas Peden[1], and Hussam Alzahrani[1]

[1]University of Otago

April 15, 2018

Abstract

A written report that details the process of deciding on a project, how long the chosen project will take to build, how we are going to build it and what platforms it will run on. An analysis of the disability supported and how the application will support that disability will also be highlighted.

Contents

List of Figures

Phase 1 - Deciding on a project

Idea one:

An application for individuals who are emotionally unstable, using some kind of artificial intelligence to provide support in times of need. After performing feasibility analysis on this idea, we decided to reject it. The primary reasons were:

- None of us are AI experts, and we don't have the time to create our own AI "chat-bot".
- Even if we could create our own "chat-bot", the relevance of its answers to a users input may vary and in some cases seem insensitive.
- If we used a cloud-based AI the trouble then becomes handling the network communication required.
- Even if things worked perfectly, this kind of application has been developed already (for example, "Replika"), and at a far higher quality than we are able to produce in such a short time frame (rep).

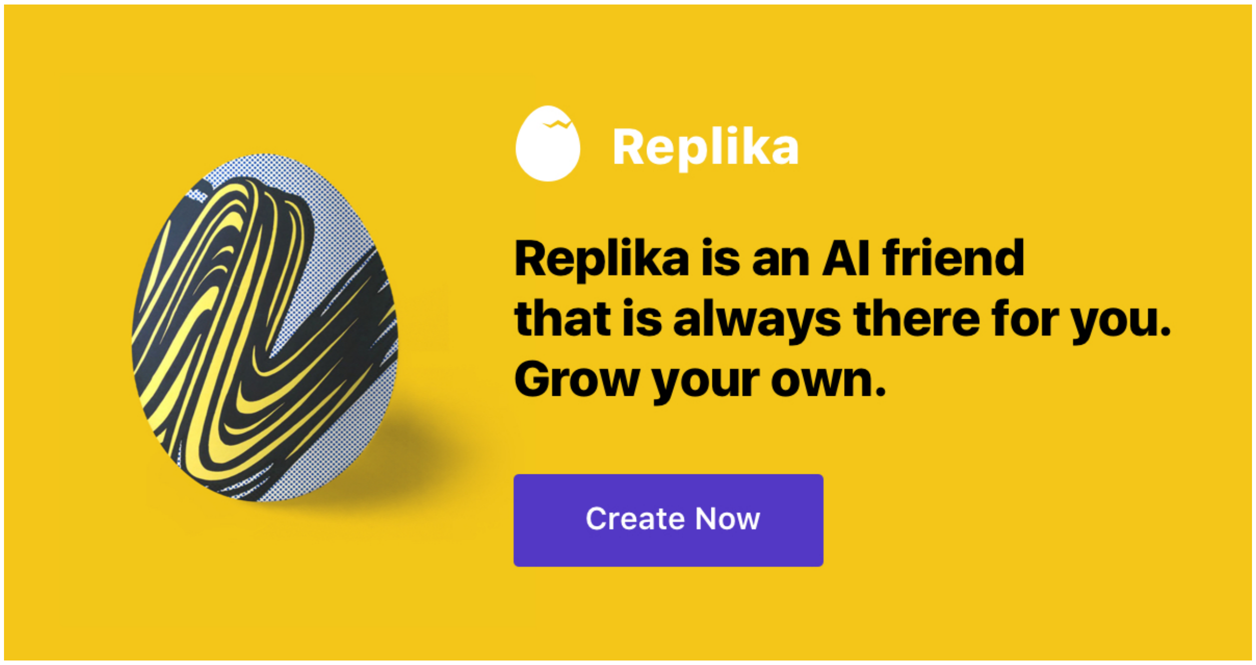

Replika
Replika is an AI friend that is always there for you. Grow your own.
Create Now

Figure 1: An Advert for Replika

Idea two:

A note taking application that would serve as a notetaker for those who cannot physically take notes, because of some kind of disability. After performing feasibility analysis we rejected this idea mainly because:

- Several types of software that achieve similar goals already exist.
- Most lecture theatres are equipped with recording equipment already.
- As we are planning on building an IOS application, this presented challenges in itself as Apple's framework for text to speech has several limitations, for example, speech recognition only lasts for around a minute at a time and Apple limits recognition per device (although the actual limit is unknown) (doc, a)

Idea three:

An application that could help a blind person identify the distance between their device and an object, recognize colors and detect the lighting in their environment by relaying this information back to the user by speech. After researching this idea we rejected it because of the following:

- Microsoft has a similar App which has more useful features such as object recognition and facial recognition.
- Most Blind people won't have a phone with two cameras on one side such as the iPhone X, reducing our target demographic significantly.
- Without two cameras it's not possible to triangulate distance unless you use the horizon line as a point of reference which is very inaccurate and would make the app unusable indoors.

Idea four:

An application for individuals who suffer from Myalgic Encephalomyelitis (ME) (aka. Chronic Fatigue Syndrome). Briefly, Myalgic Encephalomyelitis is considered a complex disability that causes cognitive defects, pain, and long periods of exhaustion(zin). We believe we can support individuals that have ME through designing an application that tracks and logs periods of exhaustion and pain, in order for health practitioners to better understand their condition. The application would allow users to enter daily information relating to pain levels, pinpointed to different areas of the body. This information would then be collated and stored persistently, allowing users to generate on the fly reports detailing the symptoms of their disability.

We settled on this idea for a few reasons:

- A group member has a family member with this condition, meaning we were able to gain insight into the disability, and how modern software could help support individuals living with the disability.
- We believe that the task is achievable within the timeframe, considering all of us need to learn Swift in order to build the application.
- The application could also extend to other individuals suffering from other conditions, and could be used to (potentially) support a range of disabilities.

Phase 2 - Refining the concept

Early on when planning how we wanted to build this application, we decided we wanted it to have two main forms of functionality. The first, would be the ability for users to tap on a life-like model and add an entry related to pain or discomfort in that particular area. The second main source of functionality would be the ability to dynamically generate a report that displayed a series of statistics. These statistics could then be sent to an email address, allowing the user to forward that data to other people, such as their doctor or healthcare professional. Various other features, such as push notifications could be used to remind users to complete their daily log. After talking with individuals who suffer from the disability, we realized it may be beneficial to have functionality that lets the user record general lethargy and pain. This allows a user to not pinpoint a specific body part if they don't want to. This extra functionality also allows the application to apply to a wider range of disabilities.

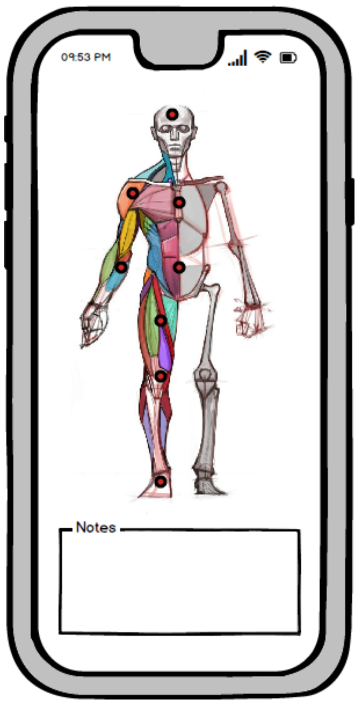


Figure 2: Early design concept

After discussing the exact details of the application, we were able to outline the main screens that will exist within the application:

- A login page where the user can sign in.
- A sign-up page where the user can sign-up if they have not got an account.
- A homepage which the user lands on after signing in.
- A user profile page that displays relevant information tied to the account.
- A summary page that displays statistics and progress based on their daily log of pain and or general lethargic periods. This page also handles the report generation so they can create a report based on the statistics they choose.

- A history page that allows the user to view previous pain logs that they had input.

It needs to be quick and intuitive for the user to input their lethargy/pain levels into the app. This is so it doesn't feel like a chore, as it would have to be done on a regular basis for the app to work as intended and give a meaningful report. Therefore the idea is to minimize the number of actions the user needs to do to input the information accurately. The app also needs to capture the subjective suffering of the user faithfully to their experience without overwhelming them with options. i.e some examples of pain include:

- Acute pain starts suddenly and is short-term.
- Chronic pain lasts for a longer period of time.
- Breakthrough pain often happens in between regular, scheduled painkillers.
- Bone pain happens when cancer is affecting a bone.
- Soft tissue pain happens when organs, muscles or tissues are damaged or inflamed.
- Nerve pain happens when a nerve is damaged.
- Referred pain is when pain from one part of your body is felt in another.
- Phantom pain is when there is a pain in a part of the body that has been removed.
- Total pain includes the emotional, social and spiritual factors that affect a person's pain experience.

The key to getting the app to work successfully is balancing the complexity of recording subjective experience accurately with a quick and easy to use interface allowing for a hassle-free user experience. *We don't want to cause the user more pain than they already suffer.*

The homepage is the first page the user will land on every day when they sign in and therefore should be the place they input their pain levels for that specific day. The page will include a picture of a human being similar to that of Fig. 2 and will be gendered in accordance with the details they signed up with. Here they will be able to record information about specific parts of the body. There will be buttons to change the figure of the human to display muscle, inner organs, bone, and skin/outer organs depending on what the user wants to record. There will also be a quick easy button to flip the human from front to back. The user can then select the type of pain classification on the side such as whether they are recording acute pain or chronic pain. The information regarding their pain classifications will be saved per organ/body part so when they come back the next day they won't have to reclassify it if they don't want to. As it is likely that the pain classification will not change day to day. When they touch a body part to record pain a simple pop up will request they record a pain rating associated with it, a simple 1 - 10 pain scale. They can do this for every body part if they wish too. There will also be options to record more general qualia that aren't directly associated with a body part. These would include options such as 1-10 scores relating to fatigue, anxiety or other qualitative states of being. The information they input will be saved to their Firebase account. The user profile will be a simple page displaying the information related to their account. It would display their name, age, gender, birth date, email and a profile picture.

The history page will allow the user to scroll through all the past logs allowing them to view what information they had entered and on what days they had entered them. The user will not be able to alter reports that are more than a couple of days old. You may have a good recollection of yesterdays pain, but altering pain ratings from a year ago could give a dishonest view of the past even while assuming good intent on the part of the user. The app would be a more honest product if it guaranteed that the user input information relating to an experience near the time of experience.

The summary page will be where the user can create a series of graphs to help visualize their pain for use either by themselves or a healthcare provider. Some potential options include:

- Percentage of total pain contributed by each body part displayed in a pie chart.
- Percentage of pain contributed by each body part for each month displayed in a bar chart.
- A body parts pain ratings over time and comparisons between multiple body parts displayed in a line graph.
- Overall fatigue scores over time.
- Average pain ratings across multiple chosen body parts.
- Feelings of mental anxiety/ stress graphed against physical pain to see if there is a correlation.
- Acute pain vs Chronic pain.

The user will be able to choose among multiple graphs to view and flick between them easily using a quick bar. Depending on the type of graph they are viewing they will have options to choose which body parts/pain options they would like to graph. If it is a type of graph that has time as an input they will be able to choose which their time span. Once they have created the graph they can add it to the report and add a comment to the graph if they so choose. After choosing a graph or adding multiple graphs to the report they can generate a pdf.

The summary page will be grabbing the data from all the logs they have previously saved to Firebase.

Who is going to build it:

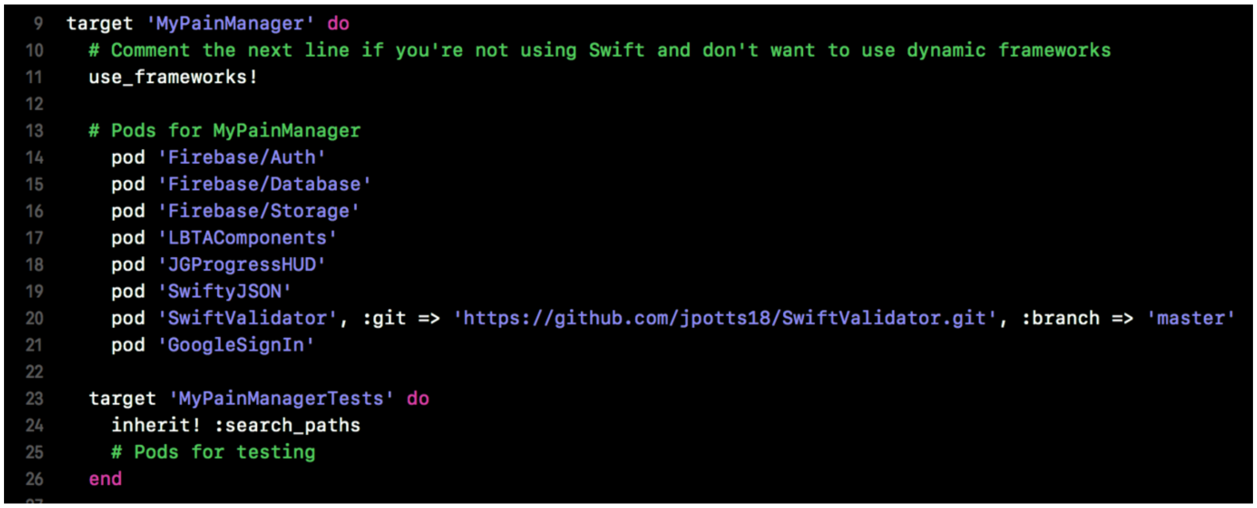
Harry Ellerm, Baz Peden, and Hussam Al-zahrani will be building the application as a team. Although some members are stronger in different areas, we are all new to Swift, so we all face the same challenges when learning how to develop an application for IOS.

How we are going to build it:

We will be developing the application for ios, making use of Swift and Xcode. In terms of a backend and user authentication, we will be making use of Google's Firebase, which (after figuring out the specifics of using it) should allow our application to be developed quickly, without having to worry about traditional database or authentication concerns, as Firebase handles a lot of the 'plumbing code' for us (fir, a). We will also be making use of third-party libraries in order to make the development process faster. Within the swift project, we plan to use CocoaPods for dependency management, which streamlines and simplifies managing external libraries (coc). Although the list of third-party plugins used will undoubtedly grow, some of them used so far include:

- Firebase Authentication (fir, b)
- Firebase Database (fir, c)
- Firebase Storage (fir, d)
- JGProgressHud (jon)
- LBTAComponents (bhl)
- SwiftyJSON (swi)

In order to identify all of the third party libraries used, you can quickly investigate the pod file, which will list all of the dependencies currently being imported.



```
target 'MyPainManager' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!

  # Pods for MyPainManager
    pod 'Firebase/Auth'
    pod 'Firebase/Database'
    pod 'Firebase/Storage'
    pod 'LBTAComponents'
    pod 'JGProgressHUD'
    pod 'SwiftyJSON'
    pod 'SwiftValidator', :git => 'https://github.com/jpotts18/SwiftValidator.git', :branch => 'master'
    pod 'GoogleSignIn'

  target 'MyPainManagerTests' do
    inherit! :search_paths
    # Pods for testing
  end
```

Figure 3: Screenshot of the Podfile as at 31/03/2018

GitHub will be used for version control (`https://github.com/HarrisonEllerm/COSC345_Software-Engineering`), and we will also be making use of various tutorials that highlight how to implement features we want to use within the project. For example, resources used so far include:

- The "Rebeloper" youtube channel (`https://www.youtube.com/channel/UCK88iDIf2V6w68WvC-k7jcg`)
- The "Let's build that app" youtube channel (`https://www.youtube.com/channel/UCuP2vJ6kRutQBfRmdcI92mA`)
- iOS 11 & Swift 4 - The Complete iOS App Development Bootcamp series (`https://www.udemy.com/ios-11-app-development-bootcamp/learn/v4/overview`)

StackOverflow will also be used when troubleshooting particular problems. We will include a link to the webpage, within the source code when we make use of a solution that has been posted on StackOverflow.

Generalized (Preliminary) UI Structure:

We will not be making use of storyboards, as we feel it may be easier to apply constraints programmatically. The main "window" of our application will be of type UIWindow, and its root view controller will be of type UITabBarController. This allows us to have multiple tabs that can be easily accessed, and can essentially drive the majority of our application (much like, for example, Facebook) (Fig. 4 and Fig. 5).

Generalized (Preliminary) Database Structure:

We will be making use of Firebase Real-time Database, which uses a JSON based representation to store data. Although we will need to adjust the structure as we are building the app a preliminary structure has been drafted (see Fig 6).

Supported Platforms:

We plan to make the application supported across all iPhones and iPads. We will focus on building the application for the iPhone initially, and then test the application on an Ipad, checking that the application functions as intended.

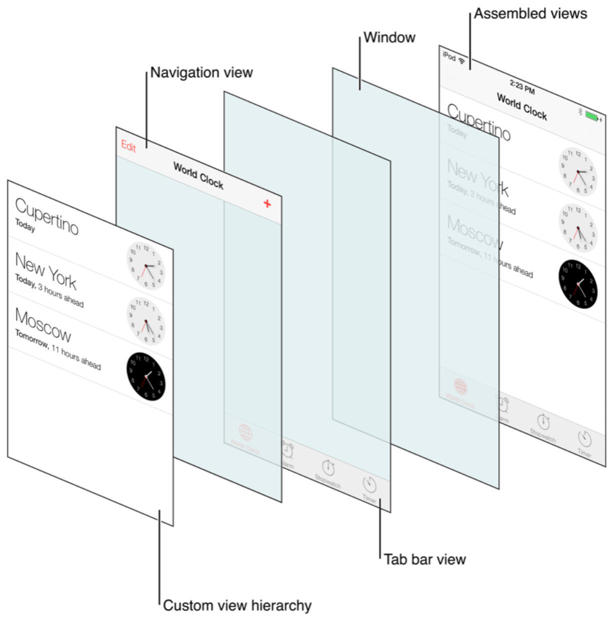
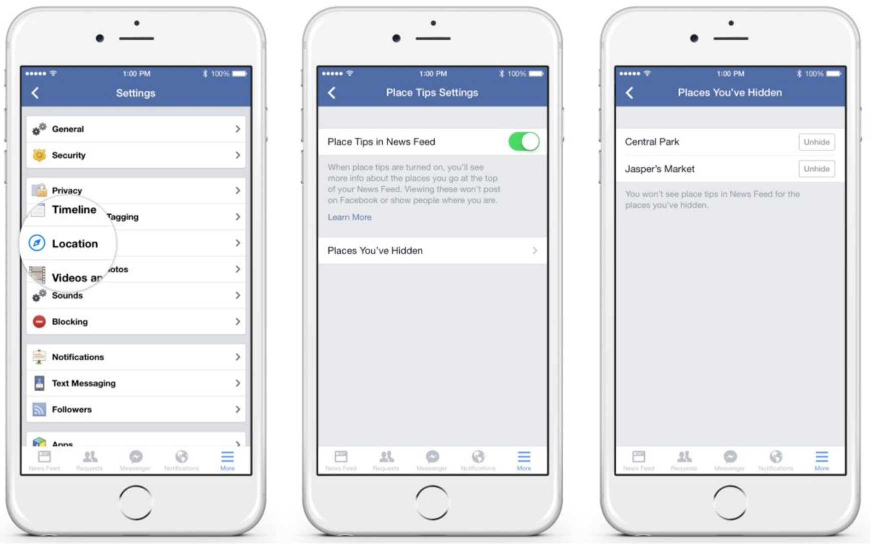


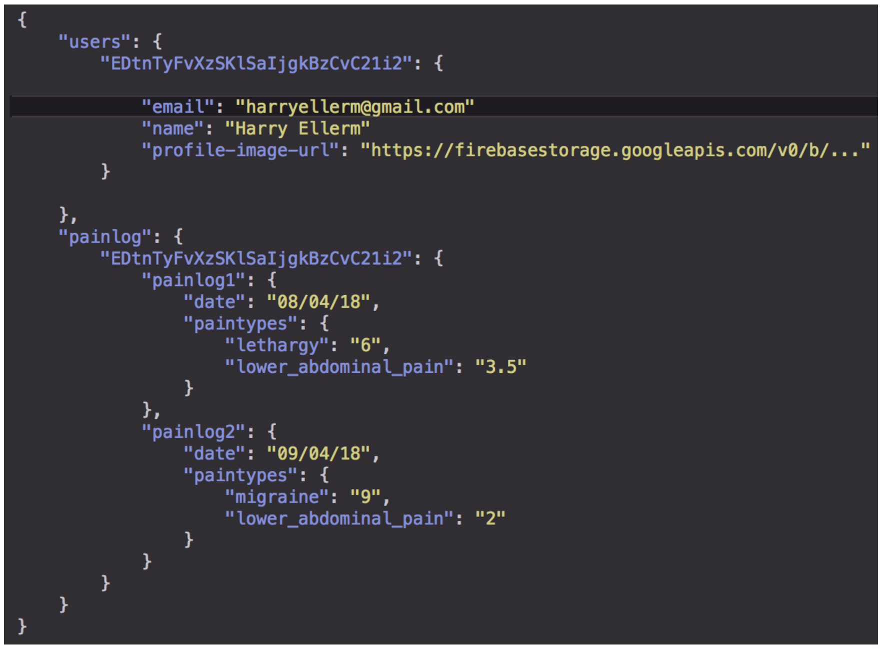
Figure 4: Example of the primary views of a tab bar controller (doc, b).



When place tips are turned on, you'll see more info about the places you go at the top of your News Feed. Viewing these won't post on Facebook or show people where you are.

You won't see place tips in News Feed for the places you've hidden.

Figure 5: The Facebook application, showing the use of a UITabBarController (bea).



```
{
    "users": {
        "EDtnTyFvXzSKlSaIjgkBzCvC21i2": {

            "email": "harryellerm@gmail.com"
            "name": "Harry Ellerm"
            "profile-image-url": "https://firebasestorage.googleapis.com/v0/b/..."
        }

    },
    "painlog": {
        "EDtnTyFvXzSKlSaIjgkBzCvC21i2": {
            "painlog1": {
                "date": "08/04/18",
                "paintypes": {
                    "lethargy": "6",
                    "lower_abdominal_pain": "3.5"
                }
            },
            "painlog2": {
                "date": "09/04/18",
                "paintypes": {
                    "migraine": "9",
                    "lower_abdominal_pain": "2"
                }
            }
        }
    }
}
```

Figure 6: Prelimary database structure example.

Phase 3 - Developing a schedule/methodology

We have developed a series of charts that outline the various responsibilities of the team, during the course of the year. This was employed as a way of reminding the team what particular areas they are responsible for, and when certain elements should be completed (see Fig.8 through Fig 10). These charts will undoubtedly change throughout time, as the project progresses. When considering a methodology, we decided to make use of an agile scrum based approach, that allows us to adapt to uncertainty and track progress (see Fig. 7). Therefore, every milestone (i.e. assignment due-date) we will be aiming to deliver a "shippable" piece of software so that the stakeholders of the application can assess its progress and identify areas for improvement.

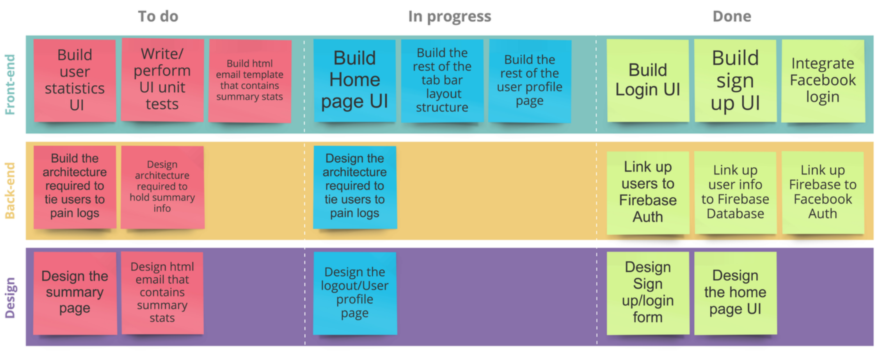

To do
In progress
Done
Front-end
Back-end
Design
Build user statistics UI
Write/ perform UI unit tests
Build html email template that contains summary stats
Build Home page UI
Build the rest of the tab bar layout structure
Build the rest of the user profile page
Build Login UI
Build sign up UI
Integrate Facebook login
Build the architecture required to tie users to pain logs
Design architecture required to hold summary info
Design the architecture required to tie users to pain logs
Link up users to Firebase Auth
Link up user info to Firebase Database
Link up Firebase to Facebook Auth
Design the summary page
Design html email that contains summary stats
Design the logout/User profile page
Design Sign up/login form
Design the home page UI

Figure 7: Current project Scrum board

First Milestone:

Within this stage, we will be writing the report, and also becoming more comfortable with a foreign programming language (Swift). As a way of applying what we are learning, we decided to try to implement the "login" and "sign-up" screen at the same time, as this takes some pressure off the next stage (Fig. 7). Although this milestone is mainly based on the report, some user stories that will be undertaken include:

Story ID	Story name	Size	Sprint	Comments
1	User Login	15	Milestone 1	"As a user, I want to be able to login into my account, so I can make use of the application." A simple screen that allows user to login to the application.
2	User Profile Page	20	Milestone 1	"As a user, I want to be able to view my profile, so that I can gain insight into the information stored in regards to my profile". A user should be able to view their profile.
4	Persistent user information/ authentication	10	Milestone 1	"As a user I want my information to be persistent, so that I do not have to re-enter it every time I use the application." Users information should be persistent, and available when logged into from any device. This involves authenticating a user via firebase, and storing their information into the database.
5	User Sign Up	5	Milestone 1	"As a user, I want to be able to sign up to use the application via email address, or Facebook, so that I can make use of the applications services". Constructing a sign-up page that is linked to Firebase, via either email or Facebook

Table 1: Milestone 1 User Stories - Size is in "story points" (one story point equals one hour).

Second Milestone:

Within this phase, we intend to get most of the UI finished and introduce baseline functionality that allows for a user to log in and log entries. This will include setting up basic chart functionality within the application to allow users to gain insights into their daily logs (Fig. 8). Testing will also be a vital part of this process and will be conducted via physical testing and making use of unit-tests within the application. Some of the user stories to be developed in this phase include:

Story ID	Story name	Size	Sprint	Comments
1	User pain log	20	Milestone 2	"As a user, I want to be able to add logs of pain and/or lethargy easily, so that I can keep track of my condition over time". A page that allows users to enter information in regards to their condition, which will be stored persistently.
2	User Summary	30	Milestone 2	"As a user, I want to be able to view summary statistics of my pain and/or lethargy logs, so that I can asses how my condition is progressing". Involves a page that contains some kind of graph that tracks their pain levels/lethargy throughout time.
3	User Email Summary	20	Milestone 2	"As a user, I want to be able to email summary statistics of my condition in a easily readable format, so that I can communicate the specifics of my condition with others". Involves a process that generates an html email of summary statistics, and sends those to a specified email address.
4	Usability	15	Milestone 2	"As a user I want the application to be responsive with a delay of no greater than a second, without understanding why the process is taking some time". Involves ensuring processes are threaded correctly, and if a process is taking a large amount of time (i.e. plus 1 second), displaying a notification to the user with some kind of HUD.
5	User Navigation	15	Milestone 2	"As a user I want to easily navigate through the application, without having to be shown how to navigate through it and access the features I want to use". Involves building out the rest of the UITabBar, and making sure all dialogs are logically presented.

Table 2: Milestone 2 User Stories

Preliminary schedule after the Second Milestone:

After we have completed Milestone 2, we will sit down and evaluate where the application is in terms of development and assess how we need to proceed. User stories will be evaluated, and (if needed) redesigned. There will no doubt be a review of the existing code base, which will most likely include some refactoring and/or redesign of certain features. Once we feel we are happy with the current application state, we will continue to work on extending functionality (this is largely improving the graphical representations, summary statistics, and report generation process).

Gantt charts outlining project progression:

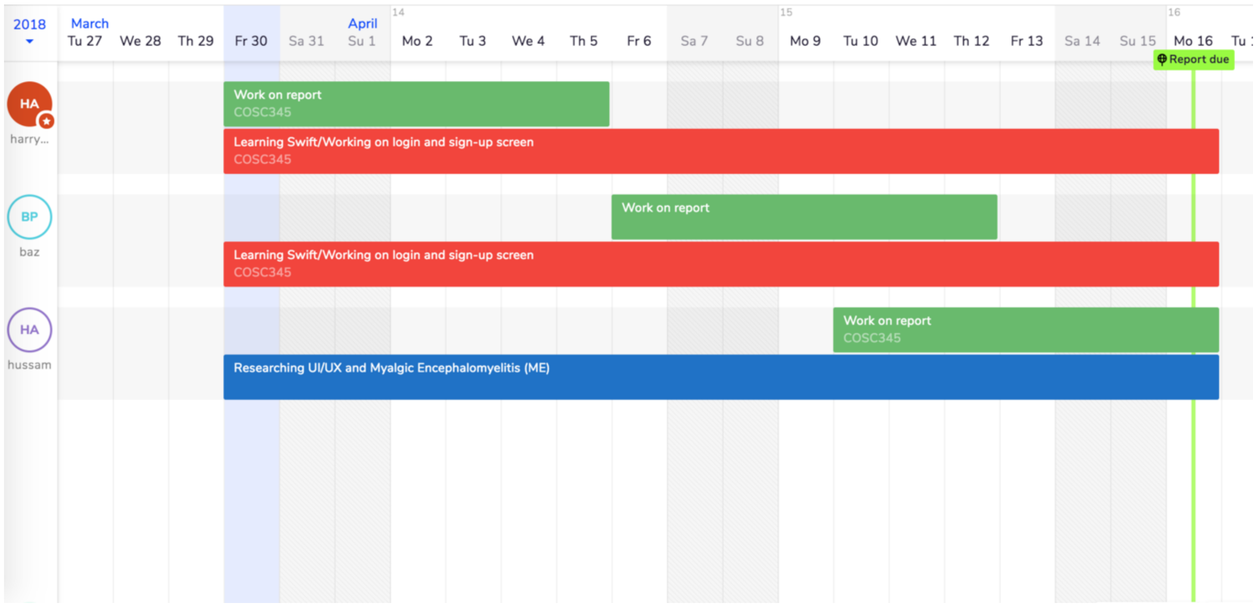
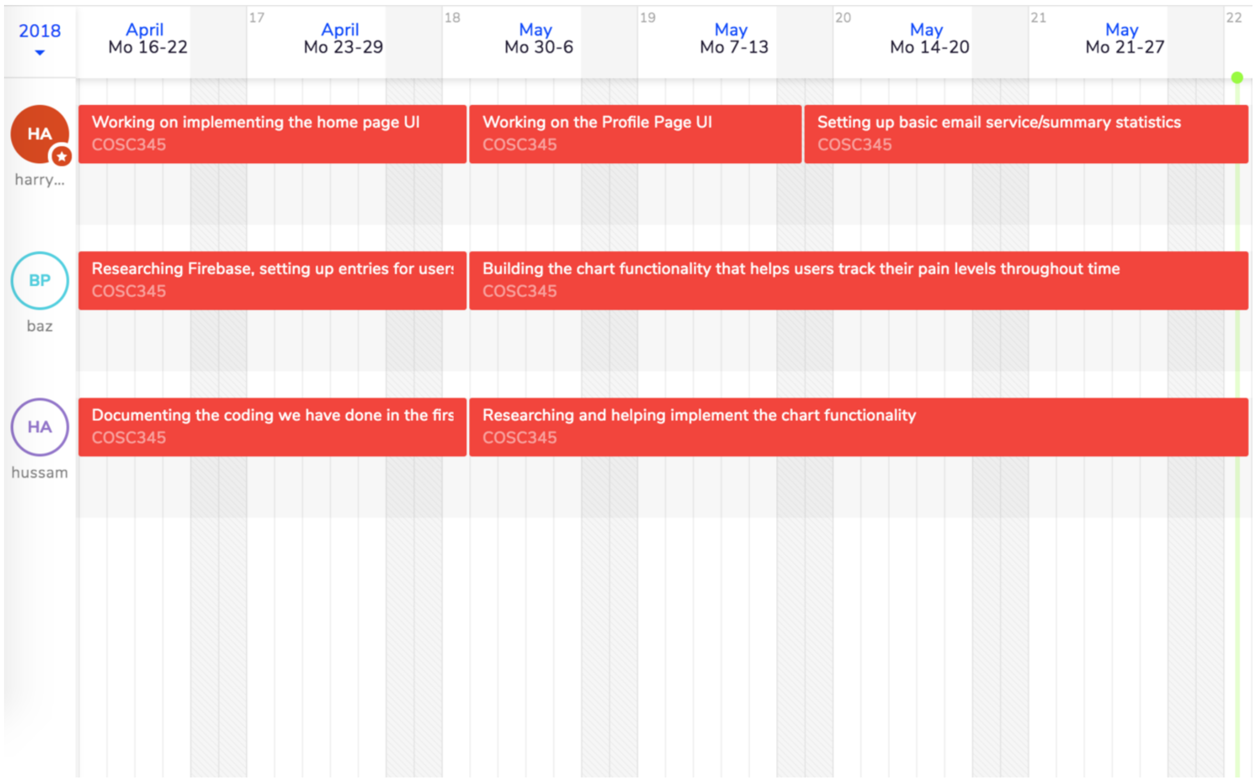


Figure 8: Phase 1 Schedule (Milestone 1 - April 16).



Working on implementing the home page UI COSC345

Working on the Profile Page UI COSC345

Setting up basic email service/summary statistics COSC345

Researching Firebase, setting up entries for users COSC345

Building the chart functionality that helps users track their pain levels throughout time COSC345

Documenting the coding we have done in the firs COSC345

Researching and helping implement the chart functionality COSC345

Figure 9: Phase two schedule (Milestone 2 - May 28th).

Phase 4 - Considering Project Risks

Throughout the project, there will undoubtedly be risks present that threaten the development of a fully functional, successfully built application. A few of the key project risks we analyzed include:

Risk no	Risk	Likeli-hood	Im-pact	Risk Exposure
1	Changes to requirments specifications when coding	0.3	8	2.4
2	Specification takes longer than expected.	0.2	6	1.2
3	Significant team member sickness affecting development.	0.5	7	3.5
4	Coding module takes longer than expected.	0.7	7	4.9
5	Testing demonstrates errors or deficiencies in design.	0.4	4	1.6
6	Group members do not communicate effectively and/or fail to work on the project.	0.2	9	1.8
7	Third party libraries in use become deprecated.	0.2	7	1.4

Table 3: Likelihood is a percentage and Impact is scaled out of 10, risk exposure calculated as Likelihood multiplied by Impact.

Risk exposure is typically considered in days. If we consider 1 day of coding to be equal to 6 hours, we can analyze the amount of time that may be lost subject to these risks. Through identification of these risks and considering their likelihood and potential impact on the project, we can rank them in terms of overall risk, via the risk exposure figure:

1. Coding Module takes longer than expected (RE = 4.9 or 29.4 hours potentially lost)
2. Significant team member sickness affecting development (RE = 3.5 or 21 hours potentially lost).
3. Changes to requirements specifications when coding (RE = 2.4 or 14.4 hours potentially lost).
4. Group members do not communicate effectively and/or fail to work on the project (RE = 1.8 or 10.8 hours potentially lost).
5. Testing demonstrates error or deficiencies in design (RE = 1.6 or 9.6 hours potentially lost).
6. Third party libraries in use become deprecated (RE = 1.4 or 8.4 hours potentially lost).
7. Specification takes longer than expected (RE = 1.2 or 7.2 hours potentially lost).

Planning for the risks identified

We will attempt to be agile in our approach to planning for project risk, as we are already taking an agile approach to project development (i.e. through the use of scrum). When researching how to plan for risks, we came across a methodology for managing risks outlined by Dr. Alan Moran in his paper titled "Agile Risk Management and Scum"(Moran 2017). He suggested that the following strategies could be employed:

1. *Do nothing and plan*: Here you accept risk but setup plans to deal with risks should they eventuate.
2. *Risk tasking*: Create a task (as part of the project schedule) that is meant to mitigate a particular risk.
3. *Risk tagging*: Mark specified risky tasks for implementation using specialized agile approaches (i.e. pair programming).
4. *Task dropping*: Removing the task altogether from the project.

We plan to make use of these techniques when assessing the risks we identified (see table 4).

Monitoring Risks

As the project progresses there will be a need to monitor existing risks and consider new risks that may appear. We will use the scrum board to keep track of these risks, as user stories are worked on. This way all team members can view risks as they evolve, and take them into account when developing a specific feature (see Fig.10).

Pain Management Application Development Best Practice

Through analysis of existing pain management application development best practice literature, we were able to gain some insight into domain specific features and/or implications that we need to consider when

Risk no	Risk	Risk plan
1	Changes to requirements specifications when coding	Create a risk task: Here we will establish a specialized task that occurs continuously throughout the project, in order to mitigate this risk. This will involve updating the project scrum board as requirements/user stories change.
2	Specification takes longer than expected.	Do nothing and plan: We will analyze the specification of the project as it develops, and organize group meetings to asses particular areas if it is taking longer than expected.
3	Significant team member sickness affecting development.	Do nothing and plan*:* Specific user stories/features will be assigned to different individuals within the group. We will also assign a “backup” developer to each of these, so that if a member falls ill, it can be picked up by another member.
4	Coding module takes longer than expected.	Create a risk tag: Here we will make use of pair programming when implementing challenging features. Although this may seem like it could slow down development, we believe it will actually increase the rate of development as it may help develop higher quality code, which could help avoid bugs/badly written code.
5	Testing demonstrates errors or deficiencies in design.	Create a risk task: Here we will establish a task that must be performed after implementing a new feature, extensive testing of the UI and code (including testing boundary cases).
6	Group members do not communicate effectively and/or fail to work on the project.	Create a risk task: We will establish weekly meetings that must be attended by all members, in order to discuss progress and challenges we are currently facing.
7	Third party libraries in use become deprecated.	Do nothing and plan*:* We sincerely hope that they won’t become deprecated. If they do, we will research how the code base will be effected, and transition the existing code to make use of the new API(s).

Table 4: Risks and associated risk plans

developing the application.

Research and Findings

Guidelines for developing healthy living apps - VicHealth, Melbourne, Australia (2015)

This comprehensive document establishes some particularly key points that may be of use:

1. ***Any development of a health-based application should involve a Health Promotion and/or Behavioural Change Specialist.***

It is suggested that health professionals play a key role in assessing whether specific algorithms will create the desired result. Understanding human behavior and how healthy living applications can seek to reinforce healthy behavior is also highlighted as being critical. Some of the critical implications this may have for our project include:

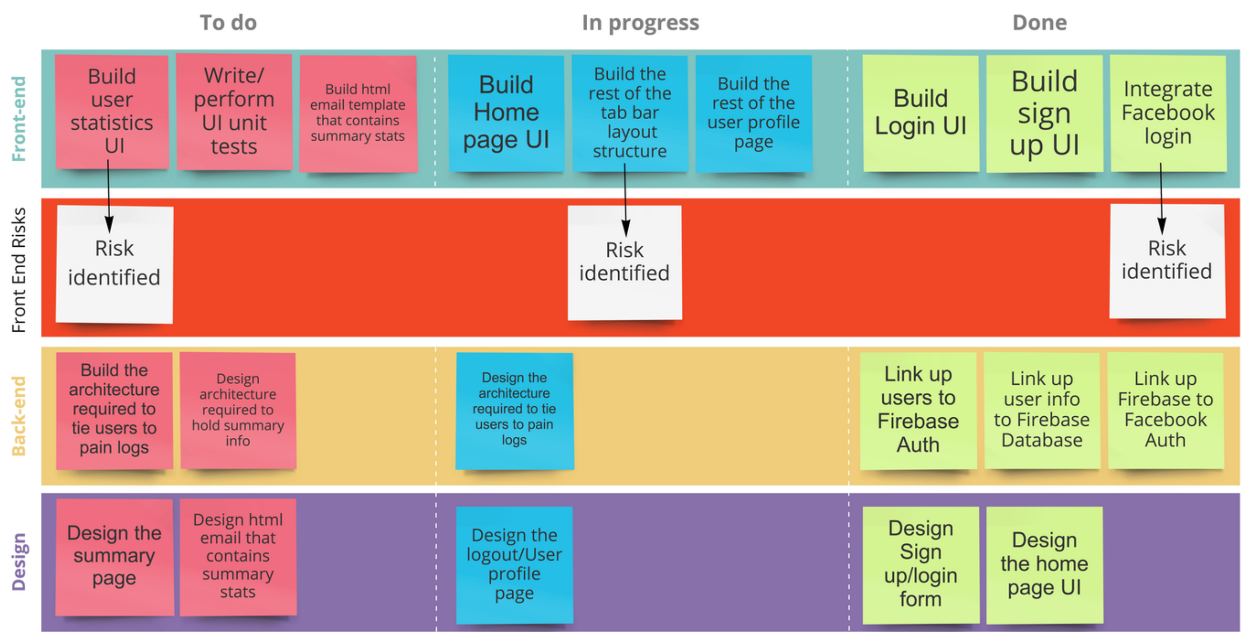

To do
In progress
Done
Front-end
Build user statistics UI
Write/ perform UI unit tests
Build html email template that contains summary stats
Build Home page UI
Build the rest of the tab bar layout structure
Build the rest of the user profile page
Build Login UI
Build sign up UI
Integrate Facebook login
Front End Risks
Risk identified
Risk identified
Risk identified
Back-end
Build the architecture required to tie users to pain logs
Design architecture required to hold summary info
Design the architecture required to tie users to pain logs
Link up users to Firebase Auth
Link up user info to Firebase Database
Link up Firebase to Facebook Auth
Design
Design the summary page
Design html email that contains summary stats
Design the logout/User profile page
Design Sign up/login form
Design the home page UI

Figure 10: Scrum board with risk column example.

- It may be necessary to contact health professionals and behavioral experts as we begin to develop the final stages of the application, that involve reinforcing positive health behavior as a way of managing chronic pain.

Action: We will assign the role of Health Promotion/Behavioural Change Specialist to Hussam Alzahrani. While we are developing and extending functionality, it will his responsibility to research the implications of this kind of functionality, and consider the impact it may have in regards to health promotion and/or behavioral change. Hussam will also establish contact with specialists in the chronic pain field if it is necessary to gain more insight into a particular area of concern within this space. A family member of one of the team is actually a General Practioner (GP), and as such may be used as a point of contact when developing specific features of the pain management application.

2. ***Involving consumers and/or other stakeholders of the application during development is key to success.***

While this is in nature an agile principal we intend to follow anyway (i.e. rapid delivery of shippable software), it is noted within the article that in the case of healthy living applications, involving stakeholders in the development of the application who could potentially be end users (i.e. stakeholders that suffer from chronic pain in our case) is vital. The implications of this on the project are:

- It is necessary to involve an individual who suffers from chronic pain in the development and testing of development iterations.

Action: We will actively be involving a family member who suffers from Chronic pain in the development of the application. We will most likely perform user acceptance testing after delivering the alpha, and at some point before delivering the beta, to ensure the application is user-friendly and useful.

3. ***Privacy and Data security are key when dealing with sensitive private health information.***

The article highlights that when developing applications within the healthy living space, individuals may often be prompted to enter sensitive information, and are at risk of their privacy being breached if this was to not be secured properly. Some of the implications of this on the application include:

- It is necessary to follow basic security measures such as password protection, adhering to health privacy principals and providing a privacy disclosure statement that outlines what information is being collected, by who, and why it is being collected.
- If ads are being used inside the application (as are normally present in "lite" versions of applications), you need to consider if these ads are invasive on the user's privacy, and what information these ads may be collecting.
- Although guidelines for minor's use of smartphone apps are yet to be established, the impact the healthy living application may have on minors should be addressed, in terms of content appropriateness and privacy.

Action: We will assign the role of security officer to Bas Peden. He will be responsible for ensuring that we are following basic security measures (to some degree this is handled for us through the use of Google's Firebase Authentication). If we eventually have the application at the stage that it is ready to be released to the app store, we will undergo a security sprint, where a privacy disclosure statement will be developed, and add use/minor's use of the application will be analyzed.

A quality review of smartphone applications for the management of pain - Portelli P, Eldred C (2016)

This article analyses some of the pitfalls application developers make when developing applications designed to aid in the management of chronic pain. Key findings highlighted:

- Current applications within the healthy living apps space rarely adhere to evidence-based guidelines.
- There is a need for developers to work with healthcare professionals in order to ensure the application actually helps those with chronic pain.
- Rigorous end-user acceptance testing and randomized control trials are essential in developing applications within the healthcare space.

Action: We will be developing this application collaboratively with end users and health professionals, through personal contacts the team has. In this way, we hope to mitigate some of the risks identified in this article.

Impact of Findings on Development

The findings of our research actually highlight a lot of important issues we had previously not considered.

- There is a large “human” factor to consider when developing an application for individuals with chronic pain. People are complex, their pain and behavior are complex, and as such, we need to have a good understanding of how the features we integrate into the application impact their wellbeing. As such we have assigned the role of Health Promotion/Behavioural change specialist to Hussam Alzahrani, who will be in charge of researching the possible implications of our features on end users, and contacting specialists outside of the group should we need to gain a deeper understanding of the issues arising.
- Users will be entering private, sensitive information into the application. As such we need to make sure our data privacy is top notch. To mitigate the risk of users data privacy being violated, we have assigned the role of security officer to Bas Peden. He will be responsible for researching Firebase Authentication, and ensuring that it will store users information securely. If we are at the stage where we are considering releasing the application to the app store, we will designate an entire two-week sprint to security implications, and develop a privacy disclosure statement.
- End-user testing is essential. We will seek to involve user acceptance testing after each scrum, to make the application is usable, as well as useful to those who suffer from chronic pain.

Conclusion

After careful deliberation and a reasoned search through the ideas space, we have decided to build an App to help people who suffer from various forms of pain related illnesses. The apps utility will enable users to document and keep track of their pain levels. We want to give our users the ability to easily communicate their experience with others in a user-friendly and visually engaging manner. The App will be geared towards but not limited to people suffering from chronic fatigue. It will enable users to log and display their pain levels over time in a way that will allow the users to communicate a subjective experience that can sometimes be difficult to otherwise convey accurately. We will be developing the majority of the App in Swift and Xcode but will also be making heavy use of googles Firebase for the back end. We plan for our App to be a cross-platform application for iPhone and iPad to meet the requirement briefing. We have set clear milestones and planned the delegation of workload between members from a macroscopic perspective across the duration of the year. On smaller time scales we will distribute the work amongst our team evenly, as our skill sets deem able and our schedules permit. Part of our plan incorporates the consideration and ongoing monitoring of risk so that we aren’t blindsided when unexpected things happen, as they most surely will. This plan enables us to gauge our performance over the course of the year, guiding us towards completing the app on time with the high level of quality we are looking to achieve.

References

Facebook launches new Place Tips feature in your News Feed, rolls out Facebook Bluetooth Beacons. http://www.idownloadblog.com/2015/01/30/facebook-place-tips/. URL http://www.idownloadblog.com/2015/01/30/facebook-place-tips/. Accessed on Sun, April 08, 2018.

bhlvoong/LBTAComponents. https://github.com/bhlvoong/LBTAComponents. URL https://github.com/bhlvoong/LBTAComponents. Accessed on Sat, March 31, 2018.

CocoaPods.org. https://cocoapods.org/. URL https://cocoapods.org/. Accessed on Sat, March 31, 2018.

Speech — Apple Developer Documentation. https://developer.apple.com/documentation/speech, a. URL https://developer.apple.com/documentation/speech. Accessed on Thu, March 22, 2018.

UITabBarController - UIKit — Apple Developer Documentation. https://developer.apple.com/documentation/uikit/uitabbar b. URL https://developer.apple.com/documentation/uikit/uitabbarcontroller. Accessed on Sun, April 08, 2018.

Firebase. https://firebase.google.com/, a. URL https://firebase.google.com/. Accessed on Sat, March 31, 2018.

Firebase Authentication — Firebase. https://firebase.google.com/docs/auth/, b. URL https://firebase.google.com/docs/auth/. Accessed on Sat, March 31, 2018.

Firebase Realtime Database — Firebase Realtime Database — Firebase. https://firebase.google.com/docs/database/, c. URL https://firebase.google.com/docs/database/. Accessed on Sat, March 31, 2018.

Cloud Storage — Firebase. https://firebase.google.com/docs/storage/, d. URL https://firebase.google.com/docs/storage/. Accessed on Sat, March 31, 2018.

JonasGessner/JGProgressHUD. https://github.com/JonasGessner/JGProgressHUD. URL https://github.com/JonasGessner/JGProgressHUD. Accessed on Sat, March 31, 2018.

Replika. https://replika.ai/. URL https://replika.ai. Accessed on Sat, March 31, 2018.

SwiftyJSON/SwiftyJSON. https://github.com/SwiftyJSON/SwiftyJSON. URL https://github.com/SwiftyJSON/SwiftyJSON. Accessed on Sat, March 31, 2018.

Myalgic Encephalomyelitis: Symptoms and Biomarkers. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4761639/. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4761639/. Accessed on Thu, March 22, 2018.