# EASE: Pain Tracker Application

## Team Details

Sam Fleury: 897306
Emilio McFadzean: 4447010
Dan Bent: 2952542
Caleb Cuttriss: 6897782

## Git Repository

www.github.com/emcfadzean/Disability-App

## Summary

Chronic pain affects more than 1.5 billion people worldwide, and is the number one cause of long-term disability in the US (Borsook, 2012; Osterweis, et al, 1987). Chronic pain disorder can affect all areas of people's lives, causing lack of sleep, anxiety, depression, as well as interfering with eating, exercise and other activities many people take for granted.

For decades doctors have been encouraging people living with chronic pain to keep daily records of their pain levels. Our app will offer a digital alternative to traditional pain-tracking methods, allowing for streamlined and personalised record-entering, as well as highly-usable data visualisation. This can empower users to take control of their pain management, and provides practitioners insight into the everyday experience of their patients.

## Chronic Pain & Pain Tracking

Chronic pain is broadly defined as pain that lasts a long time, and epidemiological studies have shown that it affects between 10% and 55% of people in various countries (Harstall C, Ospina M., 2003). Chronic pain can initially be caused by an injury, or by an ongoing illness, but often the cause is unclear. Pain is also a very personal and subjective experience, and cannot be objectively located or measured. This means that healthcare professionals rely on patients' subjective reports of the timing, location, and nature of their pain (known as their 'pain history') in order to help diagnose its cause (MedlinePlus, Spring 2011).

While chronic pain treatments often have at best mixed results, a ‘self-management’ treatment approach - in which patients take an active role in the problem-solving and decision-making around their condition - shows great promise. A central element of this approach is having the patient track their experiences in ways that allow them to make inferences about what medications and lifestyle choices work for their particular condition, and which allow for good communication with their healthcare providers (MedlinePlus, Spring 2011). It’s this kind of empowerment that we intend to facilitate with our pain tracker app.

# App Functionality and Design


Figure 1. Early concept design of home screen user interface.

## Concept

Our app will provide a digital alternative to the traditional pen-and-paper pain diary, with a focus on ease-of-use. The core functionality involves recording pain levels and symptoms at various points throughout the day. These data points are used to create a graph of “pain history”, a clear visualisation tool that can be used to make inferences over time.

### Core Functionality

We understand that if we want people to use our software everyday for a long period of time, the user experience should be as frictionless as possible. The interface should be clean and intuitive, and common tasks such as entering and reviewing records should take as few actions as possible. The primary view will feature an at-a-glance graph of records over time, a scrollable stack of cards displaying the most recent entries, as well as a floating action button to quickly record a new entry.

### Secondary Features

Moving past the core functionality, there are secondary features we'd like to include dependent on the development time required for the minimum viable product. The highest priority of these includes Google Drive integration for backup (which should be set-it-and-forget it), reminder notifications for making records and taking medications, and customisable templates for records, to allow tracking of whatever kind of data is relevant to the individual.

### Stretch Goals

Lower priority features include symptom toggles for records (to easily track occurrences of different symptoms across records), a system for recording relevant non-symptom related data (such as medication changes, exercise changes, etc.) which might be useful to visualise along with the records, weather integration in records (many chronic pain conditions are exacerbated by cold conditions), a widget to allow entering records from the home screen, a method for exporting records for medical practitioners, and deeper data visualisations.

## Development

### Research

When we were given the task of designing an app specifically with disability in mind, our first move was to find out what people need. We did this by creating a survey, which we developed and distributed with the assistance of the local branch of CCS Disability Action, a nationwide support and advocacy organisation. The results of this survey, as well as our own experiences and research led us to settle on the pain tracker concept, as well as the choice to target the Android platform. We found the people with chronic pain are more likely to have Android rather than iOS devices, due to socioeconomic disadvantages of living with disability. (A PDF of our survey questions can be found on our GitHub page.)

### Team structure and collaboration

The development work will be split up according to skills. Dan and Caleb are the more experienced coders, so they will handle the back-end development, while Sam and Emilio will oversee the design (UI/UX) tasks. This includes reporting and documentation for those related areas.

We have regular meetings twice per week, Mondays and Thursdays, and increase the frequency where required. The coordination tools we are using are Slack for ongoing communication, Google Drive for reports and questionnaires, GitHub and Android Studio for development and version control.

### Timeline

We have chosen our design carefully to ensure that once the core functionality is in place, any additional features will be minimally interdependent. This means that should we run into unexpected roadblocks or time-consuming challenges, they can be cut without undermining the basic functionality of the app.

Our alpha release is due on the 28th of May, giving us around 6 weeks from the due date of this report to get the core functionality in place and begin prototyping interfaces. A beta release is due around the end of July, by which time we will have settled on our final feature set, and be entering the testing and polish stage. In this final stage we will be finding outside testers to provide feedback on real-world use, squashing bugs, and polishing the user experience.

## Technical Specification

### Authentication

User authentication will be handled by Google Sign-In using OAuth2, allowing users to log in with their Google accounts and backup to their Google Drive. We will use a persistent database to store data locally, with support for cloud backup when an internet connection is available. This negates the need for the project to handle user accounts and the security concerns that arise with doing so. Furthermore, we know that all Android users who download our app will have a Google Sign-In, as they require one to download apps from the Google Play Store.

### Application Programming Interface (API)

The app will be built for Android and support devices running Android version 4.1.x (API level 16) and above. Additionally, the app will have added support and functionality for Android Wear devices to acknowledge and respond to push notifications from the app. However, Wear support will only be available to phones running the API version 23 and above.

### Storage and database

The persistent database will be built with the Room persistence library which offers an abstraction layer for a SQLite database. One of the benefits of this persistence library is the ability to obtain LiveData from the database using Observers. The database will also be lifecycle aware and will therefore know when to stop querying the database, based on whether the data is currently being viewed. To avoid database queries being run on the main UI thread, we will make use of RxJava for asynchronous calls. Thus, our app will be more efficient and stable.

### Security

As mentioned above, security is built into using Google Sign-in. OAuth2 offers end to end encryption, handled by Google, means that our application will not come into contact with private user account information.

### Developments tools and architecture

In order to build our app we will be using Java 8, XML and the Gradle build system. The app will be built using the Model–view–viewmodel architectural model. By utilising a ViewModel, we will be able to store and manage UI data independently. This allows for the UI components to change without the need to manage data themselves, providing a persistent container for UI data throughout the apps lifecycle.

## References

Borsook, D. 2012. *A future without chronic pain: neuroscience and clinical research.* US National Library of Medicine. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/23447793

Harstall C, Ospina M. *How Prevalent Is Chronic Pain?* June 2003 Volume XI, Issue 2 Pain Clinical Updates, International Association for the Study of Pain. Retrieved from https://www.iasp-pain.org/PublicationsNews/NewsletterIssue.aspx?ItemNumber=2136

MedlinePlus, Spring 2011. *Chronic Pain: Symptoms, Diagnosis, & Treatment* Retrieved from https://medlineplus.gov/magazine/issues/spring11/articles/spring11pg5-6.html

Marian Osterweis, Arthur Kleinman, & David Mechanic, 1987. *Pain and Disability: Clinical, Behavioral, and Public Policy Perspectives.* Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK219253/