Design implementation document

There were many important design decisions made during the implementation process of this project that we felt were important to mention. From the bottom-up perspective, and a general good software engineer tip, we decided we would have all our named constants in a <defs.h> file. This is the best way to avoid the use of “magic numbers” which are essentially random specific numbers used where no one really knows why they are there. Overall they post a vulnerability once code really gets under way, and a defs.h file that all other files can rely on can be a great way to circumvent this. Another important decision made had to do with how we choose to transport data. It was quite obvious to us upon reading the specs that a lot of data would have to get moved involving information on session time, whether the electrodes are connected and what the treatment offsets become. To achieve this in a compartmentalised manner we opted to have SessionData objects which would contain this vital info.

From there we could leverage a MVC (Model-View-Control) implementation for the rest of the code and pass along session data as needed. The M in MVC here being the Neurofeedbacksystem which hosts a lot of the typical capabilities we’d expect for our EEG device, hosting the battery and also doing all of the needed calculations. From there we have the V which falls under the MainWindow as a means to communicate information to the user in the form of the device and PC UI. From there we have the C which is the dataModel, this class is especially important for taking user interaction and communicating that to the Neurofeedbacksystem like selecting the menu options and connecting the electrodes. It also simultaneously communicates with the Neurofeedbacksystem to display important information on the ui in the form of progress changes (in the progress bar), or site specific information. This division of information is useful in terms of separating important business logic and the actual manipulation of the ui along with data that flows in between the both of them.

There are a variety of other software patterns observed, namely the singleton pattern which is typical in an MVC application. Here the dataModel will always function as a singleton as it has the means to communicate with any Neurofeedbacksystem (even multiple if we needed it to) but is singular in its cardinality as it doesn’t need to be created otherwise except to function as a controller. We also leverage other design patterns like the observer pattern where our battery power levels are being observed by several components like the dataModel and the mainWindow to ensure that it is sufficiently charged for use. Since we’re dealing with the MVC, we tried to ensure data is encapsulated when necessary, data that is not problematic being exposed (to other components, since this is contained to an application we don’t have to worry about external exposure) and more sensitive (usually process related) data is guarded and accessible through getters if necessary.

When it comes to data representation we wanted to leverage the existing QCharts in order to accomplish the display of the EEG waveforms. This ended up working well in our favour since it was relatively easy to work with and the representation looked great on the UI. We decided that the most realistic model would be that under an electrode, the frequencies change rapidly, hence the waveforms change rapidly by extension. To mimic this behaviour, we made it so the frequency that the waveform draws is randomly drawn every time the user selects a node. This resulted in the desired effect of a live feed where the frequencies of an EEG site can vary wildly (as opposed to reading the data in from a pre-set amount) allowing greater diversity in waveforms.

In conclusion, our design integrates solid engineering principles with user-centered design to create a neurofeedback device that is not only effective but also accessible to the general public. Through the careful application of design patterns and architectural decisions, the project aims to deliver a reliable and easy-to-use system that empowers users to manage their mental health independently and also represent important data in a meaningful manner.