# Design Patterns

Observer

- Qt’s signals and slots implement the observer pattern
- The observer pattern allows you to notify multiple objects about any events that happen to the object that’s being observed.

Singleton

- The singleton pattern is used to make sure that there is only one instance of the NeuresetController being created.
- The singleton is created using a protected constructor and a static getInstance() function in the NeuresetController.

Mediator

- The pattern restricts direct communications between the objects and forces them to collaborate only via a mediator object.
- All user input and selection goes through the mediator which is the NeuresetController.

# Battery Design Decisions

Overview

The Battery class is designed to simulate battery consumption reflecting typical device operation. It manages the battery level dynamically, responding to user interactions such as turning the device on or off.

Key Design Features:

1. Battery Consumption Mechanism:
- Battery consumption is triggered when the user clicks the "on" button in the UI. The battery then depletes at a rate of 10% every 15 seconds, simulating the power usage of the device during operation.

2. Automatic Shutdown
- When the battery level falls below 20%, a visual warning is displayed in the UI to alert the user. If the battery level reaches 0%, the device automatically turns off.

3. QTimer for Time Management:

- A QTimer is utilized to manage the timing of battery consumption. This timer starts when the "on" button is clicked and stops when the "off" button is pressed.

4. Signal and Slot Mechanism:
- The battery level updates and low battery warnings are handled through Qt's signal and slot mechanism, facilitating communication between the Battery class and the UI.

5. Integration with MainWindow:
- Description: The Battery instance resides within the MainWindow class, integrating closely with the application's main control flow simplifying interactions between different components of the application and enhancing cohesion.

6. Turning on when the battery is depleted
- When the battery is depleted and the device is turned off, subsequent attempts to turn it on will result in it staying on briefly before turning off again, unless recharged. This behavior is designed to prevent misuse and simulate the need for recharging before further use.

## Treatment Design Decisions

Overview

Treatment of the 21 EEG Sites starts automatically when the user selects 'NEW SESSION' from the menu.

Process:

1. The currently selected waveform band (alpha, beta, delta, or theta) in the GUI is considered the current brain state of the patient when treatment begins.

2. A treatment-delivered signal with a value of false is sent to the main window to illuminate the green treatment light.

3. The baseline of the specified waveform type is calculated using the EEG site's 'calculateBaseline()' function.

4. Treatment occurs in 4, 15-second rounds. In each round, the offset frequency is increased by 5hz.

5. Each of the 21 EEG sites are treated by adding some value in the range of +/- the offset frequency to the baseline of the waveform. This process of adding the offset frequency happens 16 times over 1 second.

6. Treatment is then completed and a treatment-delivered signal with a value of true is sent to the main window so that the new baselines for each EEG site are saved to the session logs.

## Logging of treatment sessions

Key design Features:
1. When a treatment session has completed the time and date of that session along with the before and after baselines are saved to a file and printed to the console.
2. A signal is sent to the main window when treatment is completed.
3. The slot in the main window saves the session log to a file (SessionLog.txt) and prints it to the console.
4. If a session is not completed the incomplete session data is not saved when "Session Log" in the menu is clicked it prints the information from all the sessions that have been done.

## Green, red and blue light for device status

Key design Features:
Green Light
1. The green light indicates a treatment running, it turns on once the treatments starts
2. A signal is sent to the main window from the controller saying that the treatment has started.
3. If the treatment is paused the green light turns off
4. Once a treatment is resumed the green light will turn back on.
5. Once treatment is complete a signal is sent to the main window saying that the treatment is complete and the green lights will turn off.

Blue Light
1. The blue light turns on when a new session is started
2. If the session is paused the blue will stay on to show that a session is in progress.
3. Once treatment is complete a signal is sent to the main window saying that the treatment is complete and the blue lights will turn off.

Red Light
1. The red light indicates that contact with one of the EEG sites has been lost.

2. Once connection is reestablished the light turns back off.
3. More information provide in “Connection loss between electrodes and the device”

## Connection loss between electrodes and the device

Key design Features:
1. The user can select specific EEG sites to lose contact using the spinner box.
2. By pressing the “Lose Contact” button the selected site will lose contact.
3. Any number of EEG sites can lose contact (from 1-21).
4. When contact is lost a signal tells the controller that contact has been lost on some EEG site.
5. The controller signals the EEG site that contact was lost on that specific site.
6. A signal from the EEG site tells the controller that a site has been disconnected and to pause the treatment.
7. The slot in the controller then sends a signal to the main window saying that at least one site has been disconnected and to turn on the red light.

## Connection reconnect between electrodes and the device

Key design Features:
1. By pressing the “Reconnect” button all disconnected sites will be reconnected.
2. When contact to be reconnected a signal tells the controller that contact is to be reestablished.
3. The controller signals the EEG site that contact is to be reconnected.
4. The EEG site reconnected the sites that have been disconnected.
5. A signal from the EEG site tells the controller that a site has been reconnected and to resume the treatment.
6. The slot in the controller then sends a signal to the main window saying that all EEG sites have been reconnected and to turn off the red light.

## Timer Design Decisions:

Key design Features:
1. To align with the mediator design pattern, the timer for the treatment progress was implemented inside the NeuresetController class. The NeuresetController class as a mediator will control the activity of the whole treatment, including the timer.
2. Once a new session is started, the timer will start its countdown, until the treatment is done.
3. All actions regarding the timer(s) need to be triggered by the user/GUI, and hence, the controller. For example, if the user clicks the “pause” button, the timer

and the treatment will pause at the same time, and if the user clicks the "play" button, the treatment will resume.

4. If the user pauses over 5 minutes or if the connection is lost over 5 minutes, the session will automatically end and the timer will be reset. But for testing purposes, we change the 5 minutes to 5 seconds. To handle this, we have a "handlePauseTimeout()" slot to receive the signal from the default timer class.
5. Signals are used for the interaction between mainWindow Class and the controller class, such as, timeUpdated() signal updates the time displayed in the GUI. updateProgressBar(int) will update the progressBar in the mainWindow and reset() will reset everything to default settings. Therefore, we make good use of signals and slots to build the connections between classes, namely the controller and the mainWindow class for GUI interaction.
6. It is also worth mentioning that we have another timer which is for "treatmentRound". If there is no timer to control the treatment, all treatments will be completed at once, in total of 4rounds which is not the expected and ideal behaviour. Therefore, another timer to count for each round of the treatment was added and each round will need 15 seconds.

## Calculating the Baseline Design Decision

Key design Features:

1. The uniqueness of each value in EEG sensors was an important key decision in simulating brainwave frequencies. That is why rand() functionality is used to generate brainwave frequencies.
2. The baseline is been calculated by getting the average of the brainwave frequencies. Each EEG site does its own calculations separate from one another to focus on the area it is treating.
3. 60 seconds was the decided treatment length, so the baseline is recalibrated every second of a minute. That way, we can deliver the most precise treatment to the patient.

## Calculating the Band Values

Key design Features:

1. The uniqueness of each value in the band was an important key decision in simulating the correct frequency pattern. That is why rand() functionality is used with limitations of the given band's frequency width.
2. The band values are calculated by getting the mod of the difference between the lowest and highest bandwidth frequency and then adding the value of the starting bandwidth frequency to calculate random values of the band frequency.