

Traceability Matrix

ID	Requirement	Related Use Case	Implemented by	Tested by	Design
1	User starts a new session	Normal Operation (Use case 1)	Mainwindow::startSession(), MainWindow::electrodeConnectionCheck()	Choosing the "New Session" option in the menu after turning on all EEG's	The startSession() first checks if all the EEGs are turned on and if the device's battery is less than 10%. If any of these requirements are not met, the device does not start a new session and prints an error to the console. The startSession function also keeps a sessionInProgress boolean that helps stop the session whenever the session is paused / ends.
2	Timer begins once contact is initiated	Normal Operation (Use case 1)	Session::startTreatment(), Session::Session()	Viewing the elapsed time after a session is completed.	A QTimer currentTime is initialized in the session constructor. Once the playSession() function is called to resume or start a new session, the QTimer is turned on and keeps track of the elapsed time.
3	Device calculates the overall baseline for all 21 EEG sites concurrently at the start of the session.	Normal Operation (Use case 1)	Session::startSession(), Session::calculateBaselineAvg(), EEG::calculateDominantFrequency()	Choosing the "New Session" option in the menu after turning on all EEG's	The startSession() function is called at the beginning of a session, and the calculateBaselineAvg() is used to calculate the start averages. These start averages are stored in an array "startAverages". The calculateBaselineAvg() function calculates the baseline average by looping through each eeg site and calling calculateDominantFrequency()

					quency(). This function iterates over the frequencies and amplitudes in the waveform. Each pair of the frequencies and amplitudes are then placed into a formula to calculate the baseline.
4	Device calculates the baseline average frequency for each EEG site before treatment (1 minute)	Normal Operation (Use case 1)	Session::startSession(), Session::calculateBaselineAvg()	Viewing the session Logs after a treatment is completed	Baseline average frequency is calculated for each eeg site using the calculateBaselineAvg() function. This function calculates the baseline average by looping through each eeg site and calling calculateDominantFrequency(). This function iterates over the frequencies and amplitudes in the waveform. Each pair of the frequencies and amplitudes are then placed into a formula to calculate the baseline.
5	Treatment starts, adding an offset frequency of 5hz to the baseline and recalculating every 1/16th of a	Normal Operation (Use case 1)	Session::startTreatment(), Session::recalculateBrainwaveFrequency(), EEG::calculateDominantFrequency(), EEG::applyTreatment()	Choosing the "New Session" option in the menu after turning on all EEG's, console outputs treatment progress	When a new session is started, startTreatment() is called. startTreatment() constantly checks for connection loss from the EEG sites. If there are no connection errors, treatment is applied to each EEG site by passing in an offset into the applyTreatment() function. The applyTreatment() function simply

	second, 16 times (1 second)				increments the baseline based on the offset. This offset is changed depending on the current round.
6	Device calculates the baseline average frequency for each EEG site after treatment	Normal Operation (Use case 1)	Session::stopSession(), Session::calculateBaselineAvg(), EEG::calculateDominantFrequency()	Viewing the session Logs after a treatment is completed	The stopSession() function is called at the end of a session, and the calculateBaselineAvg() is used to calculate the end averages. These end averages are stored in an array "endAverages". The calculateBaselineAvg() function calculates the baseline average by looping through each eeg site and calling calculateDominantFrequency(). This function iterates over the frequencies and amplitudes in the waveform. Each pair of the frequencies and amplitudes are then placed into a formula to calculate the baseline.
7	Green light flashes during treatment	Normal Operation (Use case 1)	Session::startTreatment(), Session::startRound(), Session::greenLightOn(), Session::greenLightOff(), MainWindow::treatmentLedHandler()	Choosing the "New Session" option in the menu after turning on all EEG's, Green light is flashed during recalculations	The MainWindow contains a treatmentLedHandler() function that is connected to a signal in the Session class called turnOnGreen. This signal is controlled by the greenLightOn() and greenLightOff() functions in the Session class. They are used at the start of a treatment, and at every round of adding an offset frequency.

					Once the treatmentLedHandler() receives the signal, it changes the style sheet's background color to green.
8	Blue light is turned on when timer is initiated	Normal Operation (Use case 1)	MainWindow::electrodeConnectionCheck(), MainWindow::contactLedHandler()	Choosing the "New Session" option in the menu after turning on all EEG's	The MainWindow contains a contactLedHandler() function that changes the style sheet's background color to blue. It is called one time inside the electrodeConnectionCheck() function. It turns on a blue light once all nodes are connected, indicating the start of the timer and a new session.
9	Red light flashes when contact is lost	Connection loss between electrodes and device (Use case 4)	Session::checkIfConnectionLost(), MainWindow::contactLedHandler(), turnOnRed()	Disconnecting and EEG during treatment	The MainWindow contains a lostLedHandler() function that is connected to a signal in the Session class called turnOnRed. This signal is controlled inside the checkIfConnectionLost() function. The function emits the signal if one of the EEG's is disconnected. Once the lostLedHandler receives the signal, it changes the style

					sheet's background color to red.
10	If after 5 seconds contact is not reestablished, the session is terminated and the device turns off automatically	Connection loss between electrodes and device (Use case 4)	Session::checkIfConnectionLost(), MainWindow::contactLedHandler(), turnOnRed(), Session::stopSession(), Session::pauseSession()	Disconnecting and EEG during treatment, then waiting 5 seconds to see the device turning off. / Pausing the session, then waiting 5 seconds to see the device turning off	The session is paused when the user clicks the pause button or when an eeg node is disconnected during treatment. The stopSession() function is called in both these cases. It starts a timer for 5 seconds and if the session is not unpaused / the eeg node is not reconnected the session is terminated and the device turns off automatically.
11	Timer shows approx time remaining	N/A	MainWindow::updateCountDown()	Choosing the "New Session" option in the menu after turning on all EEG's, timer should be visible on the device	Once a new Session is started, the ui element for the countdown timer is shown on the device, The countdown timer then constantly updates the ui element to show the integer "time left". The "time left integer starts at 35 and is decremented every second until it reaches 0 and the session is completed.
12	Session progress bar is indicated by a percentage	N/A	MainWindow::updateProgressBar(), MainWindow::updateCountDown()	Choosing the "New Session" option in the menu after turning on all EEG's,	Once a new Session is started, a ui element QProgressBar is shown on the device and is set to a range of (0, 35). The progress bar is constantly updated with the help of the "timeLeft"

				progress bar should be visible on the device	integer to show progress on the treatment. The “time left integer starts at 35 and is decremented every second until it reaches 0 and the session is completed.
13	Session log preview history that displays the time, date and session name	Therapy history viewing (Use case 2)	MainWindow::deviceLogsPreview()	Selecting the “Preview Logs” option in the “Session logs menu” then viewing the information on the device itself	The MainWindow contains a deviceLogsPreview() function that is called if the user selects the Preview session logs option. This function gets information from the sessionsLog list and formats it to fit into a QStringList. That list is then passed into an updateMenu function that displays the session logs onto the device.
14	Upload to pc option that allows user to view before and after baselines for each session	Therapy history viewing (Use case 2)	MainWindow::writeLogsToFile(); MainWindow::loadLogsBrowser();	Selecting the “Upload To Pc” option in the “Session logs menu” then viewing the baseline information on the PC ui.	After a session has been successfully completed, the user can select the “Upload To Pc” option in the menu. When this is selected, the writeLogsToFile() opens up a new file for writing and prints out session log information for each session. The loadLogsBrowser() then reads the logs from the files and copies the entire html from that file onto the ui QTextBrowser element called “PC”.
15	Option to	N/A	MainWindow::setTime(), MainWindow::setDate(),	Selecting the “Time	If the “Change Date” or “Change Time” options

	change current date and time		displayCurrentDateAndTime()	& Date” option in the menu then adjusting the time and date	are selected, the mainWindow makes a ui element visible, this ui element then allows the user to change the date or time respectively. It then updates the current time/date using the setTime() and setDate() functions provided in the QDateTimeEdit object.
16	User can press pause voluntarily during a session	Connection loss between electrodes and device (Use case 4)	MainWindow::pauseSession()	Clicking the pause button during the session	The sessions can be paused by pressing the QPushButton “pauseButton”. This button is connected to the pauseSession() function that sets the boolean sessionInProgress to false, and pauses the current Session.
17	Device displays a “battery low” message when device battery reaches 20%	Battery low response (Use case 3)	MainWindow::drainBattery()	BatteryLow QDialog appears when device battery reaches 20%	When the device reaches 20%, the drainBattery() function creates a batteryLowMessage Object. batteryLow.exec() is then called which displays a QDialog window indicating that the battery is low.