Requirements Traceability Matrix

ID	Requirements	Use Cases	Implemented by	Tested by
1	The menu has a power button to turn on the EEG device and turn it off.	#1 #10	Device::Device() bool Device::getPowered() void Device::togglePower() Device::~Device()	Tested using device.c and the mainwindow.ui “POWER” button.
2	The menu has three options: new session, session log, and a date and time settings.	#2 #3 #3a #3b	Device::Device() void Device::startSession() string Device::getTime() void Device::setDateTime(string t) string Device::getRecords()	Tested using device.c and the mainwindow.ui “New Session”, “Time and Date” and “Session Log” buttons.
3	Pressing the new session option opens a timer that begins once contact is initiated, indicated by the blue light on the device.	#3 #6	void Device::startSession()	Tested using device.c and the mainwindow.ui “New Session” button and the top left light boxes.
4	If contact is lost, the red-light flashes, the session is paused, and the device starts beeping until contact is reestablished.	#6a	int Device::getDisconnectTimer() void Device::applyTherapy()	Tested using device.c and the mainwindow.ui the top left light boxes.
5	If contact is not reestablished after 5 minutes, the device turns off automatically and the session is erased.	#7-9a #7-9a1	int Device::getRunningTimer() void Device::stop()	Tested using device.c and the mainwindow.ui “Disconnect Sensors” button.
6	The timer shows approx. time remaining and session progress bar indicated by a percentage.	#7-9b2	string Device::getTime() bool Device::getRunning()	Tested using device.c and the mainwindow.ui progress bar and timer window
7	The user can press pause voluntarily during a session.	#7-9b	void Device::pause() bool Device::getPaused()	Tested using device.c and the mainwindow.ui “PAUSE” button.
8	The same rule applies, if after 5 minutes contact is not reestablished, the session is terminated, and the device turns off automatically.	#7-9b1	void Device::applyTherapy()	Tested using device.c and the mainwindow.ui “Disconnect Sensors” and “Connect Sensors” buttons.

9	The software calculates a baseline for each EEG site individually over approx. 1 minute, determining the average dominant frequency for that site then applies the treatment over the duration of one second.	#7 #8 #9	void Device::startSession() void Device::applyTherapy()	Tested using device.c and the mainwindow.ui "Advance" button and progress bar interface.
10	During that second, the green light flashes indicating treatment is being delivered. It then moves on to the next site.	#7 #8 #9	void Device::applyTherapy()	Tested using device.c and the mainwindow.ui "Advance" button and the top left light boxes.
11	At the end of the session, a baseline is once again calculated for all 21 EEG sites.	#9	void Device::applyTherapy() string Device::getRecords()	Tested using device.c and the mainwindow.ui "Session Log" button.
12	The menu also has a session log history. Pressing this button displays the time and date of the sessions and the user can scroll through them.	#3a	string Device::getRecords() string Record::printDevice() string Record::printPC()	Tested using device.c and the mainwindow.ui "Session Log" button.
13	The before and after baselines are recorded and can be uploaded to a PC with the date and time log information. The baselines show the before and after dominant average frequencies for each EEG site, taken during the overall baselines at the beginning and end of the session, compared side by side as a numerical value.	#2a #3a	string Device::getRecords() bool Device::getPCConnection() void Device::togglePCConnection() void Record::setBeforeSignal(int current, int before) void Record::setAfterSignal(int current, int after) void Record::setRecord(string time) void Patient::setSignal(int signal, int signalChange)	Tested using device.c and the mainwindow.ui "Toggle PC Connection", and "Session Log" button. Can be viewed in the PC view window or the Device Window depending on whether the PC is connected "Toggle PC Connection".
14	The third menu option is simply a date and time setting. The user inputs the current date and time so the device clock can accurately track the sessions.	#3b	string Device::getTime() void Device::setDateTime(string t)	Tested using device.c and the mainwindow.ui "Time and Date" button and the date and time window box to change it.