Traceability matrix

ID	Requirement	Related use case	Fulfilled by	Test	description
1	Power: “Turn On/Turn Off” and “Ending A Session”	UC-01 UC-02	MainWindow	Open the UI and test the power button for the on/off functionality. With “Ending A Session” follow the steps from use case 1.	Users can turn the device on or off using the power button by holding it down. Soft will activate when a session is over, or by holding the power button during the session.
2	Battery Level	UC-03	MainWindow	Open the UI and turn on the device, and check if the battery level is shown. You can also edit the battery level in admin to see how the bar graph changes. Also select the longest session, and check if the 2 bars at the bottom start blinking when the battery is too low.	When the device is turned on, the battery level is shown on the bar graph. Also, during a session, the bar graph will periodically show battery level. Finally, if the battery gets too low, the bottom 2 bars on the graph will blink.
3	Selecting a session	UC-01	MainWindow	Open the UI and execute the steps from use case 1, and verify everything works properly.	When the device is on by clicking the power button will be able to switch between session times, and switch between types using the up and down buttons.

4	Connection test	UC-05	MainWindow	Turn on the device, and get to the point where you select a session, and verify the bar graph is showing a connection. You can also change the connection level on admin to see if the bar graph changes.	While the device is on, and you are about to select a session the bar graph will show the current connection level by lighting up certain parts of it. 8-7 is a bad connection, 6-4 is an okay connection, and 3-1 is a good connection.
5	Intensity	UC-01	MainWindow	Turn on the device and perform the steps to have a session selected. At that point, verify that you can use the up and down arrows to increase/ decrease the intensity. Also check when changing the intensity that the bar graph displays the current intensity value.	When you have selected a session, you should be able to use the up and down arrows to increase or decrease the intensity for the session.
6	Recording	UC-04	MainWindow, db	Before starting a session, select true for “Record Session” in admin. When the session is completed, verify its data was saved to the Recorded Session History section in admin.	This feature when selected saves the data of a session to a db, where the user can then view the session history of sessions that were recorded. They can view the history in the admin section.

Use cases

UC-01	Base Use Case
Description	Use case for normal use of the device
Actors	User, Battery
Pre-Condition	Device is off and there is enough battery to power on the device
Post-Condition	The device works according to the users commands
Main Sequence	1. User holds down the power button to turn on the device 2. Connect ear clips 3. User selects time duration 4. User selects session 5. User can increase or decrease the intensity as they like 6. Soft-off will trigger after the intensity and set, and it will slowly decrease the intensity over time 7. Once the session is over, the device will automatically turn off
Variation	1. If the battery is too low, it will blink 2 bars advising the user to change the battery
Extension	1. The user can hold the power button to end a session early and power off the device 2. If the battery does not last until the session is over, it will also automatically turn off the device 3. If the aux cord is not plugged in, it will turn off both of the ear plugs 4. The battery will blink when the battery level reaches 20% and below to warn the user that the battery is getting low
Design Decisions	The abstraction given to us via QT C++ means we can use the built-in UI object as an observer. Any changes made to the device can be detected in real time via the ui object.

	This removes any need for getters and setters functions and makes separate classes redundant. The UI object also functions as a mediator between the database and the controller/logic of the code. We don’t need to access the contents directly from the database class, and instead can access the elements from the QComboBox. We used this pattern when populating the session history drop down menu in our admin panel. We also made widespread use of the state pattern design. We had variables that would track whether the battery level graph, intensity slider, or the connection level should be displayed, mainly with booleans. Lastly, we also relied on the command design pattern heavily. Each button on the UI/admin panel will execute its own code that it can only access.

UC-02	Low Battery Use Case
Description	Use case for force shutdown of the device
Actors	User, Battery
Pre-Condition	Device is off, Battery has insufficient power
Post-Condition	Device is off
Main Sequence	1. User presses and holds the power button to turn off the device 2. Current sessions ends 3. Device shuts down
Variation	1. Not enough battery level for required session
Design Decisions	Upon the user pressing and holding the power button, trying to turn on the device, there is a function called powerOn that

	checks the battery level via the Qt UI object afforded to us. If the battery level is 0, the device will not continue unless the battery level is increased and the user pressed and holds the power button once more.

UC-03	No Session Selected Use Case
Description	Use case for force shutdown of the device when the user does not select a session within 2 minutes (30 seconds in the sim)
Actors	User, Battery, Device
Pre-Condition	Device is off, Battery has insufficient power
Post-Condition	Device is off
Main Sequence	1. User powers on the device 2. Device has sufficient power, powers on 3. User does not select a session 4. Device powers off after 30 seconds
Variation	1. Not enough battery level for required session 2. User may select a session and time, but does not confirm the selection
Design Decisions	When the user first turns on the device, a QTimer begins counting down from 30 seconds. If the user fails to select and confirm a session in time, the program calls a turnOff() function, which turns off the entire device and resets the variables used in the program back to their original state.

UC-04	Battery Depletion Use Case
Description	Use case for when battery is depleted when

	using the device
Actors	User, Battery
Pre-Condition	Device is on, user selected session and intensity
Post-Condition	Device is off
Main Sequence	1. User selects time duration 2. User started a session 3. Battery depletes in a fixed rate 4. When Battery level reaches 20% it will start blinking warning the user that it only has 20% battery left 5. Battery reaches 0% 6. Device shuts down
Variation	1. User may increase the battery during via the admin panel
Design Decisions	There are several helpers that monitor the battery level of the device, if it reaches levels below 20%, a helper will alternate between 2 png images using a QTimer. If the battery level were to ever each 0, the program will output text saying the device is shutting down and a turnOff() function, which turns off the entire device and resets the variables used in the program back to their original state. The user can increase or decrease the battery level through the admin panel. The battery degradation is dynamic. It is based off of the intensity and connection level of the device as per the device specifications.

UC-05-1	Recording Use Case
Description	Use Case for recording on the device
Actors	User, Recording device
Pre-Condition	Device is on, user is selecting session and session time length. CES and ear clips are attached, connection level is good.

Post-Condition	Session has started
Main Sequence	1. User selects time duration 2. User started a session 3. User chooses to record a therapy 4. It is added to history of treatment
Variation	1. If there is no recording device, you will not be able to record a session
Design Decisions	There is a QComboBox labelled “Recording” that contains a True or False option. If True is selected, the program will take the user's settings that are taken right before the device is powered off via Soft Off. If the user powers off the device manually, we considered that not regular use case and the user “disrupting” the functionality of the product. The database values are held completely in memory as we did not have information regarding the external recording module, so we made an assumption that recordings are not stored on the device. The database will write the data to a QComboBox as aQList and the strings are parsed as QStrings and ints based on the index in the QList. This allows for the UI to act as a mediator and allows for quick and easy overriding of the other settings, sessionType, sessionTime, intensity, which in turns changes the GUI without any helpers or setters.

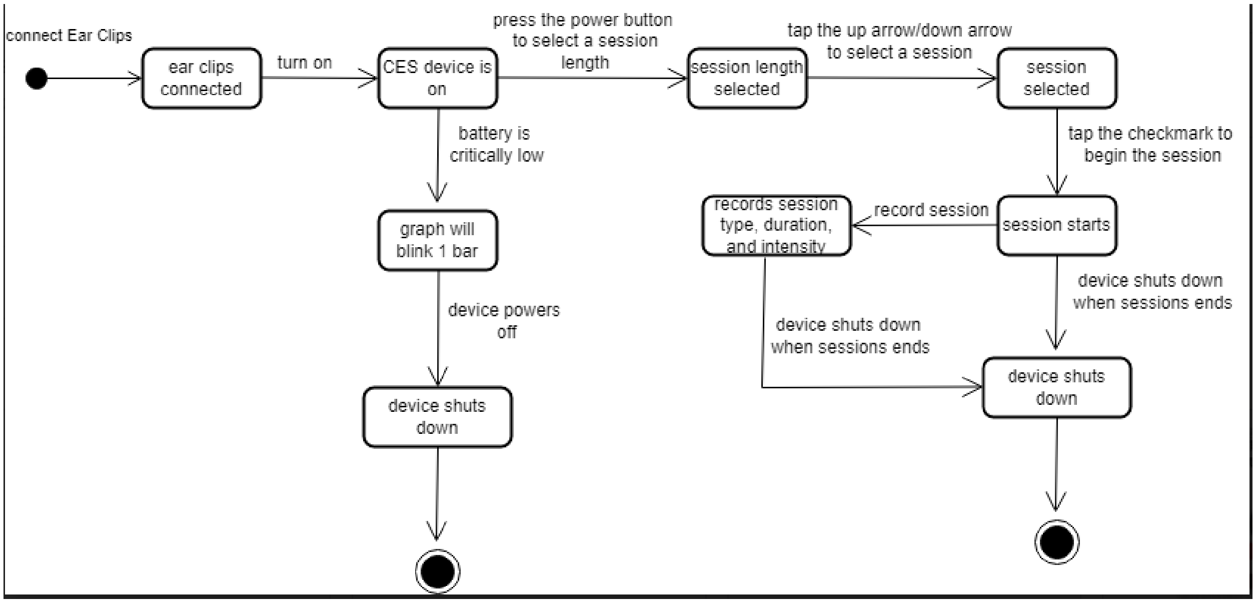
UC-05-2	Recording Sub Use Case
Description	Use Case for recording on the device
Actors	User, Recording device
Pre-Condition	Device is on, user is selecting session and session time length. CES and ear clips are attached, connection level is good. User has already recorded a session.

Post-Condition	Session has started
Main Sequence	1. User, via the admin panel, selects one of their previously recorded sessions 2. Session, time, and intensity settings are overridden 3. User confirms the selection
Variation	2. If there is no recording device, you will not be able to record a session 3. User can edit the intensity during the session
Design Decisions	See UC-05-1 Recording Use Case: Design Decisions

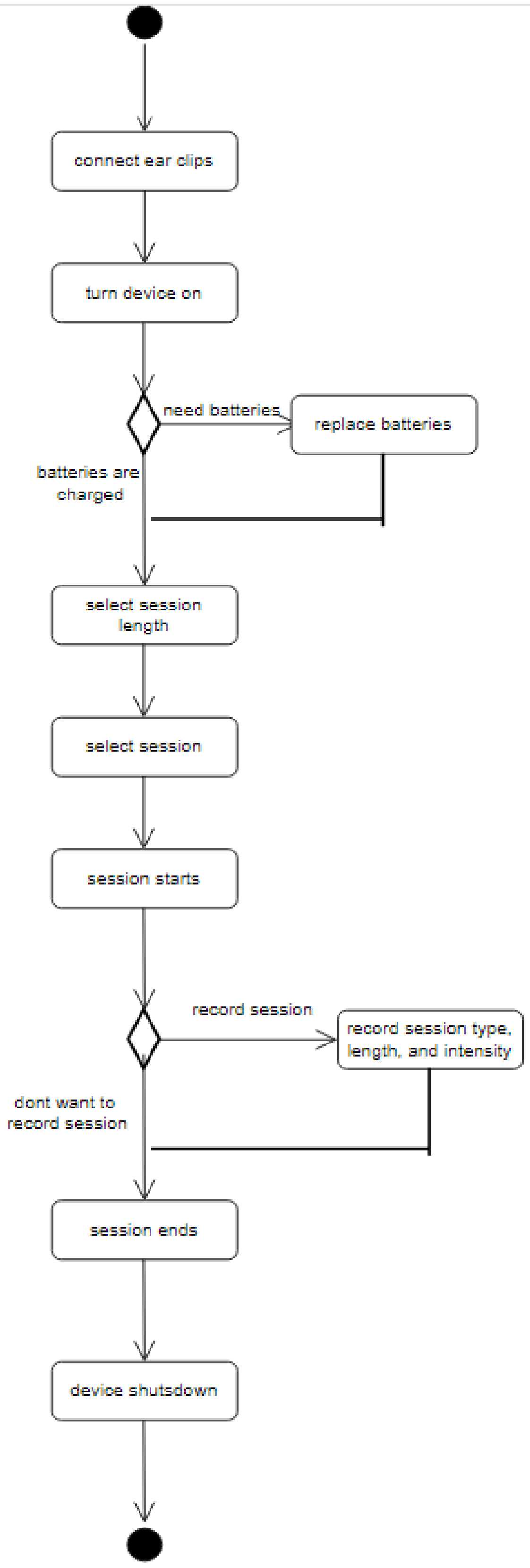
UC-06	Bad Network Use Case
Description	Use Case for Bad Network
Actors	User, Network, Device
Pre-Condition	Device is on
Post-Condition	User can select session time and type
Main Sequence	1. Hold Power Button to turn on the device 2. CES cable is connected 3. Left ear clip is connected to the user 4. Right ear clip is connected to the user 5. Connection level can be set to good or okay via the admin panel 6. User can select a session
Variation	1. User can disconnect the ear clip or CES connection during a session to pause it 2. User can disconnect the ear clip or CES connection during selection process, pausing the selection
Design Decisions	Whether the clips are connected or not, CES included, is done via the values of their respective QComboBoxes. If the CES connection is false, the user cannot set either ear clip values to true. If the CES connection does not terminate to the machine, then the machine has no way of

	telling if the ear clips are connected, so the assumption is made that they are not connected. Additionally, the user can not set the CES clips to be connected if the device is not powered on, as the device has no power to tell if the CES is connected. Furthermore, the connection will always be overridden as “Bad Connection” if the CES or ear clips are not connected and the “Bad Connection” can not be changed by the user. Monitoring the connection via the QComboBox/UI allows us to tell if any of them have been disconnected during a session. Allowing us to pause the session with just additional parameters in our if statements allowing the program to continue the session/softOff(), etc.

State diagram


connect Ear Clips
ear clips connected
turn on
CES device is on
press the power button to select a session length
session length selected
tap the up arrow/down arrow to select a session
session selected
battery is critically low
graph will blink 1 bar
device powers off
device shuts down
tap the checkmark to begin the session
session starts
record session
records session type, duration, and intensity
device shuts down when sessions ends
device shuts down when sessions ends
device shuts down

Activity diagram


connect ear clips
turn device on
need batteries
replace batteries
batteries are charged
select session length
select session
session starts
record session
record session type, length, and intensity
dont want to record session
session ends
device shutsdown