COMP 3004B

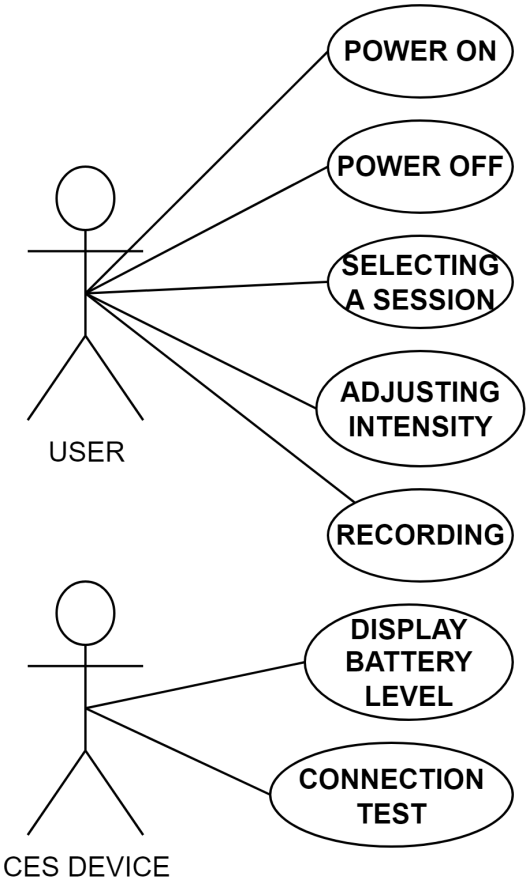
Team Project: Developing and Testing a CES Device Simulator

Israel Babalola, Sean Chung, Richard Kim, and Erica Morgan
101147765, 101156263, 101138475, 101015691

April 17, 2022

Index

1. Use Case Model



USE CASE 1: Power ON

Primary Actor: User

Scope: Oasis Pro System

Level: User Goal

Stakeholders and Interests:

User - wants to turn On the device and end a session

Battery - wants to make sure the battery is sufficient to turn the CES Device on

CES Device - wants to make sure power on functionality works

Precondition: 9V battery is present

Minimal Guarantee: None

` Success Guarantee: User successfully turns on the device.

Main Success Scenario:

1. User presses the power button.
2. LED lights turn on
3. Graph displays the battery level.

Extensions:

2a. LED light does not turn on

2a1. The 9V Battery was dead.

USE CASE 2: Power OFF

Primary Actor: User

Scope: Oasis Pro System

Level: User Goal

Stakeholders and Interests:

User - wants to turn off the device and end a session

CES Device - wants to make sure power off functionality works

Precondition: CES Device is on and is in the middle of a session, or is not in a session.

Minimal Guarantee: None

Success Guarantee: User successfully turns off the device and is able to end a session at will

Main Success Scenario:

1. User presses the power button.
2. Device turns off.

Extensions:

1a. User does not press the power button

1a1. If the user did not select a session for 2 minutes, turn off the device.

1a2. If the session has ended, enter Soft Off state, and turn off the device.

3a. Device does not turn off

3a1. User did not press the power button

USE CASE 3: DISPLAY BATTERY LEVEL

Primary Actor: CES Device

Scope: Oasis Pro System

Level: User Goal

Stakeholders and Interests:

User - wants to be notified of the current battery level

CES Device - wants to provide safe treatment for the user.

Precondition: CES Device is on and 9V battery is inserted.

Minimal Guarantee: None

Success Guarantee: Battery level is displayed properly, notifying the user when the battery level is low or critically low.

Main Success Scenario:

1. Display the battery level of the CES Device properly for a couple of seconds.

2. Make sure the battery is monitored and displayed to the screen periodically while the sessions are running.

Variation:

1a. If the battery is low, display 2 bars and blink the LED lights.

1b. If the battery is critically low, display a single bar and blink the LED light.

2a. If battery level is critically low while a session is running, end the session prematurely and signal the battery level indicator.

Extensions:

1a. Warning to replace battery is shown

1a1. Battery is critically low after turning on the device

USE CASE 4: SELECTING A SESSION

Primary Actor: User

Scope: Oasis Pro System

Level: User Goal

Stakeholders and Interests:

User - wants to select the appropriate session for their treatment.

CES Device - wants to provide safe treatment for the user.

Precondition: Use Case 1: Power On and then CES Device has displayed the initial battery level.

Minimal Guarantee: None

Success Guarantee: Treatment selected by the user successfully initiates.

Main Success Scenario:

1. User presses/releases the power button to switch between groups.
2. Chosen group icon is changed.
3. Press the INT up or down buttons to highlight a session number.
4. The frequency and mode icons associated with the session light up to indicate what frequency range and CES pulse type will be used.
5. Press the select button to start the highlighted session.
6. The session number will flash and the session will begin after a 5 second delay.

Extensions:

3a. No sessions are available in a group,

3a1. No numbers will be lit.

USE CASE 5: CONNECTION TEST

Primary Actor: CES Device

Scope: Oasis Pro System

Level: User Goal

Stakeholders and Interests:

User - wants to ensure “Excellent Connection” with the device.

CES Device - wants to provide safe treatment for the user.

Precondition: Use Case 4: Selecting a Session, session has been initiated by the user.

Minimal Guarantee: CES Device will blink, indicating “No Connection”.

Success Guarantee: CES Device confirms the connection is “Okay” or “Excellent”.

Main Success Scenario:

1. The user connects the ear clips to their ear lobes.
2. CES Mode Light will blink to indicate the device has entered test mode.
3. The graph will display the status of the connection.
4. The display will go blank.

Variation:

4. The display shows a Soft On animation.

Extensions:

3a. Okay Connection is indicated

3a1. Wet your earlobes with tap water or a tiny amount of conductive gel for better connection.

3a2. Increase intensity

3b. No connection is made during the connection test

USE CASE 6: Adjusting Intensity

Primary Actor: User

Scope: Oasis Pro System

Level: Subfunction

Stakeholders and Interests:

User - wants to select the appropriate intensity for the session.

CES Device - wants to provide safe treatment for the user.

Precondition: CES Device connection test completed.

Minimal Guarantee: None

Success Guarantee: Intensity level acts in accordance with the users specification.

Main Success Scenario:

1. User presses the INT up/down button.
2. The topmost lit number will blink from the graph 1 to 8.
3. The user will continue adjusting until the stimulation can just barely be felt.

Extensions:

3a. May result in skin irritation.

3a1. Intensity was set too high.

USE CASE 7: Recording

Primary Actor: User

Scope: Oasis Pro System

Level: User Goal

Stakeholders and Interests:

User - wants to record their session and view the history.

CES Device - wants to make sure the sessions get recorded and displayed correctly.

Precondition: CES Device is on and has completed a session.

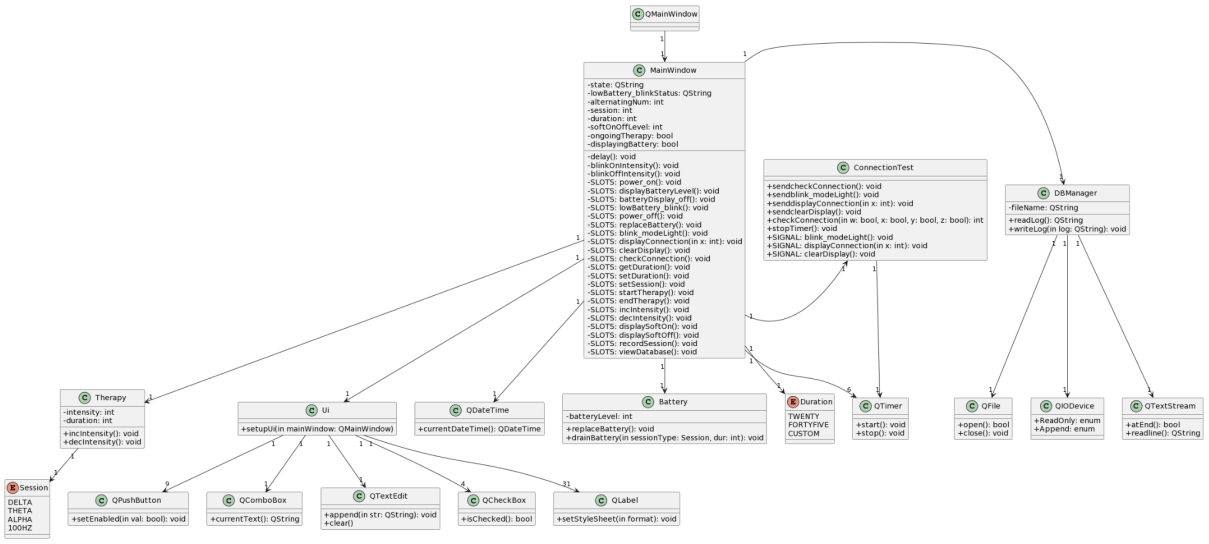
Minimal Guarantee: None.

Success Guarantee: Session is recorded properly with the correct session type, duration and intensity level.

Main Success Scenario:

1. The user presses the record button.
2. CES Device keeps track of session type, duration and intensity level of the latest session.
3. Session is added to the database properly.

2. UML Class Diagram and Design Justification



Note that a PNG file of the UML is included along with this document for easier viewing.

Few assumptions had to be made / points to note for the UML Class Diagram:

1. Getters / setters / constructors / destructors are not shown.

2. Objects are not directly written, as they are shown by the relationship of the classes.

3. I did hear a TA mentioning that the prof stated we should only include important instance variables, but I found the ‘Introduction_to_UML2’ pdf on Brightspace suggesting that we should include non-important ones as well. I was also not taught from COMP 2404 to only include the important ones. Although I do agree that it will be more efficient to not include some of the instance variables, I also think there is value in knowing the unimportant instance variables as well, as they might give one idea on how they may use those variables in future updates of the program if needed.

4. Few assumptions had to be made regarding classes from Qt. MainWindow was assumed to be a child of QMainWindow and so the diagram shows QMainWindow having one instance of the MainWindow. QTimer is shown in the diagram, showing that MainWindow uses 5 of them as objects. From the default code from Qt, it seemed that Ui is a separate class, and so it was indicated to be a separate class, owned by MainWindow. Ui will of course contain 9 QPushButton objects, 1 QComboBox object, 1 QTextEdit object, 4 QCheckBox objects, and 31 QLabel objects used on GUI, which are shown in the diagram as well.

5. My formatting generally took the method I used in my Assignment 2, since I did not lose any marks for it, I am assuming it is right.

6. It can be seen that this project uses the Mediator pattern, and uses the MainWindow class as the ESC (AI) from Elevators. MainWindow class is the main class that brings all the other classes together and works as the main computing class for the project. MainWindow class contains an object of:

    Therapy Class - used to set the therapy session

    Ui Class - adds GUI to the project

    DBManager Class - used to record session data and view the records. The records are saved in a text file.

    QDateTime Class - used to get current date and time for recording sessions. This class was not added as an object of the DBManager Class since we thought it made sense for the main part of the device to keep track of the date / time, in case we later want to implement a clock on the CES device.

    Battery Class - used to keep track of the battery level, and implement battery replacement.

    Duration Class - used to set enums for therapy duration names.

    QTimer Class - MainWindow requires 5 timers:

        batteryLevelTimer, powerOffTimer, batteryLow_blinkTimer, sessionTimer, softOnOffTimer.

    ConnectionTest Class - the class that handles functionality regarding testing connectivity as each therapy session initiates. Since ConnectionTest Class needs to access the Ui, it includes SIGNALS that are sent to SLOTS in MainWindow.

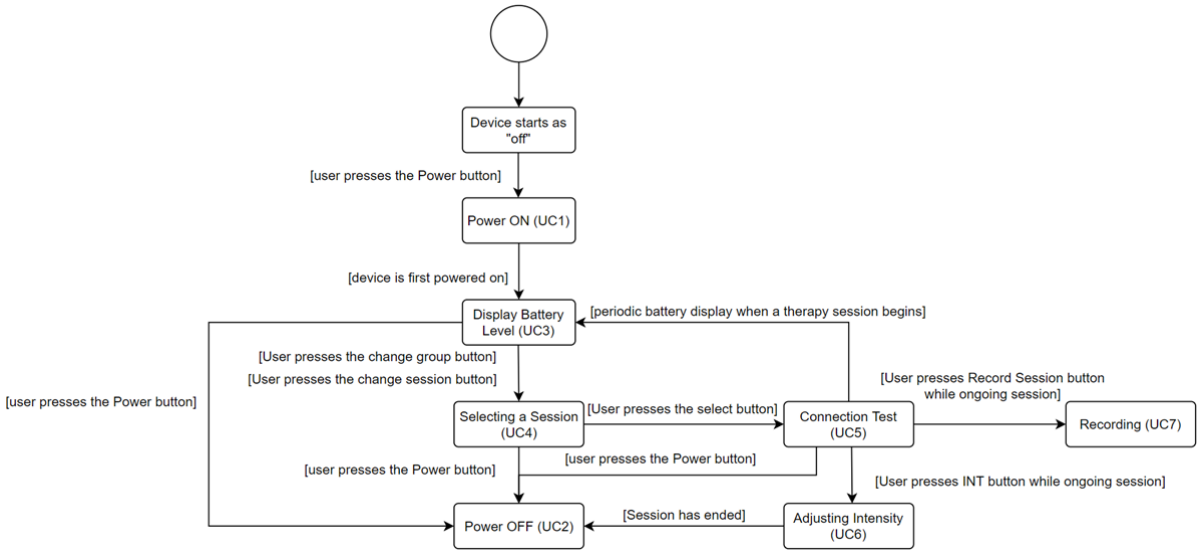
7. More detailed explanations of each class can be found in the comments of the code.

8. SLOTS and SIGNALS are used to access the Ui, and they have been stated in the UML for easier identification.

3. State Diagram

This is a state diagram that shows the relationship between all of the use cases. It shows how each use case is triggered from the CES Device.

Note that a PNG file of the diagram is included along with this document for easier viewing.

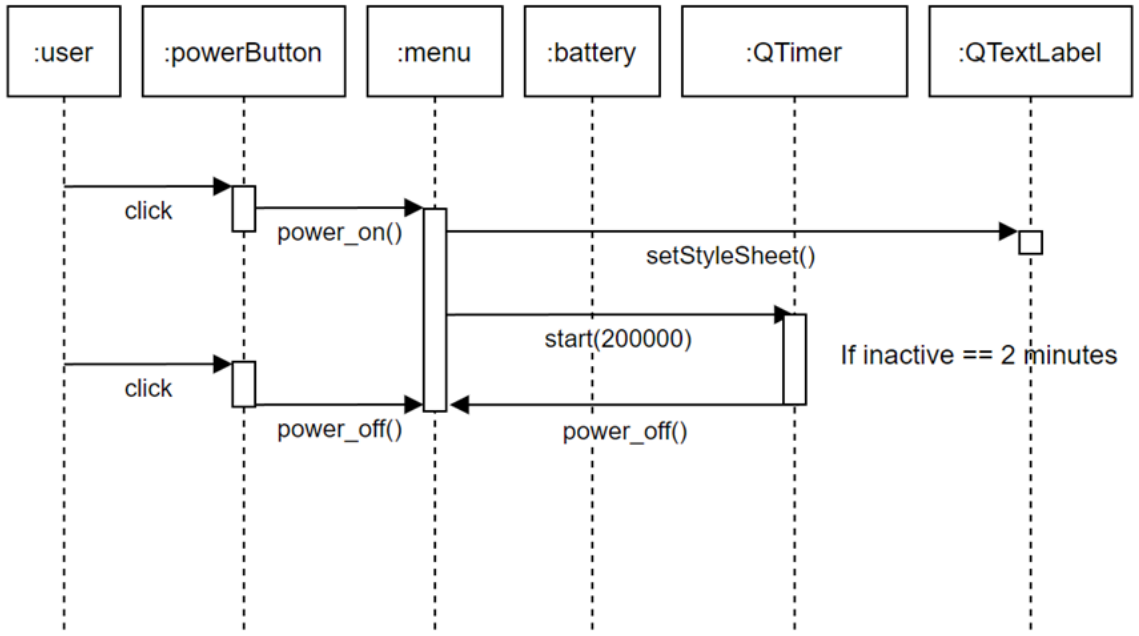


4. Sequence Diagram

Note that a PNG file of the diagrams are included along with this document for easier viewing.

POWER ON/OFF

- The first user interaction depicted is for showing the sequence of actions following a click of the power button while the CES device is off. A signal-slot relationship will trigger the ‘power_on’ function to be called inside of the MainWindow (Menu), where it will make the appropriate changes to the stylesheets (for color) of the QTextLabel.
- After the device is turned on, a QTimer will start for 2 minutes. If a session is not started within 2 minutes, the QTimer will call the power_off function. Also, if the user presses the power button at any time when the device is on, it will also signal the power_off function to be called.

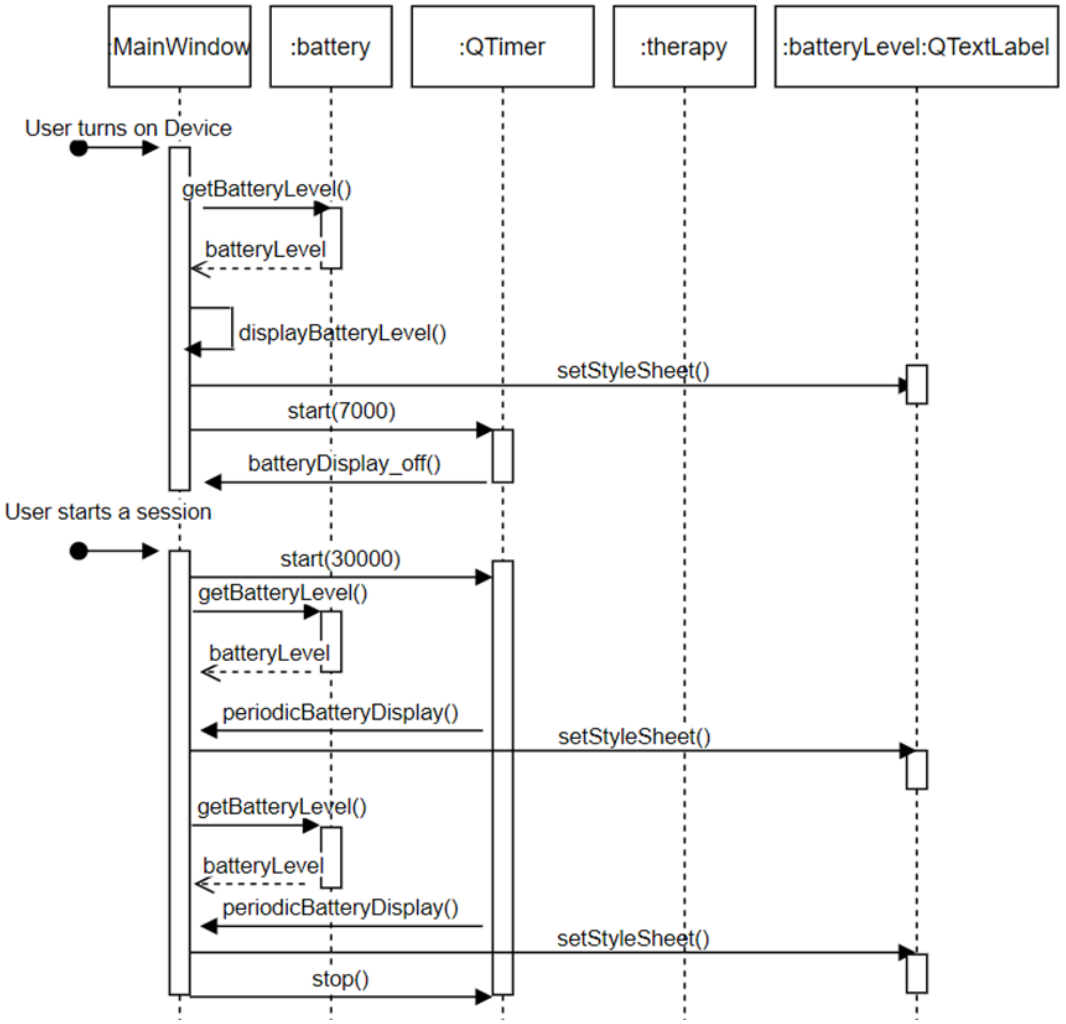


DISPLAY BATTERY LEVEL

In the Oasis Pro Manual, there were ways the battery level could be displayed on the CES Device.

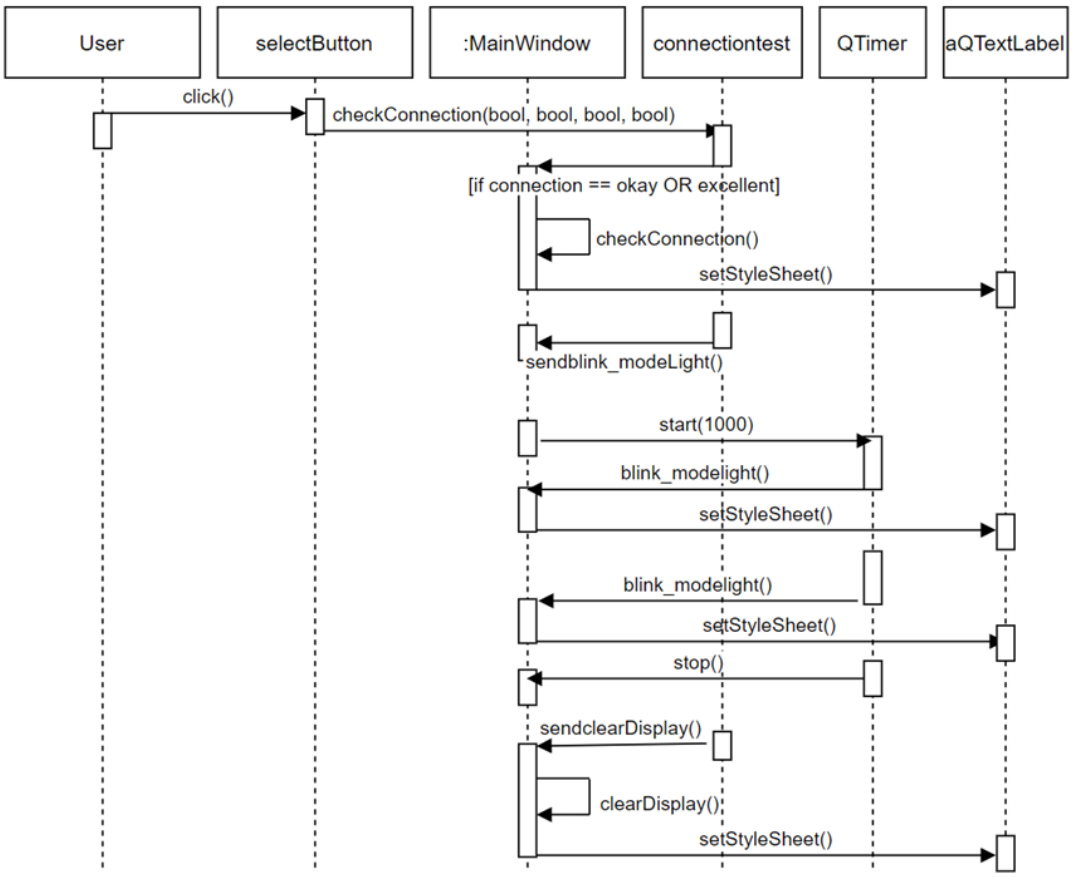
- Firstly, when the user first turns on the device, the MainWindow (Menu) will retrieve the current battery level for the CES Device, and display it by changing the colours of the QTextLabels (through setStyleSheet) that represent the battery levels. Once the battery level is displayed, a QTimer will start for 7 seconds. After 7 seconds, it will call the 'batteryDisplay_off()' function.

- Secondly, when the user starts a session, the manual says to display the battery periodically while the session is ongoing. While showing this in the sequence diagram, we did not implement this in our code. The CES Device will only display the battery level when the CES Device is first turned on.



CONNECTION TEST

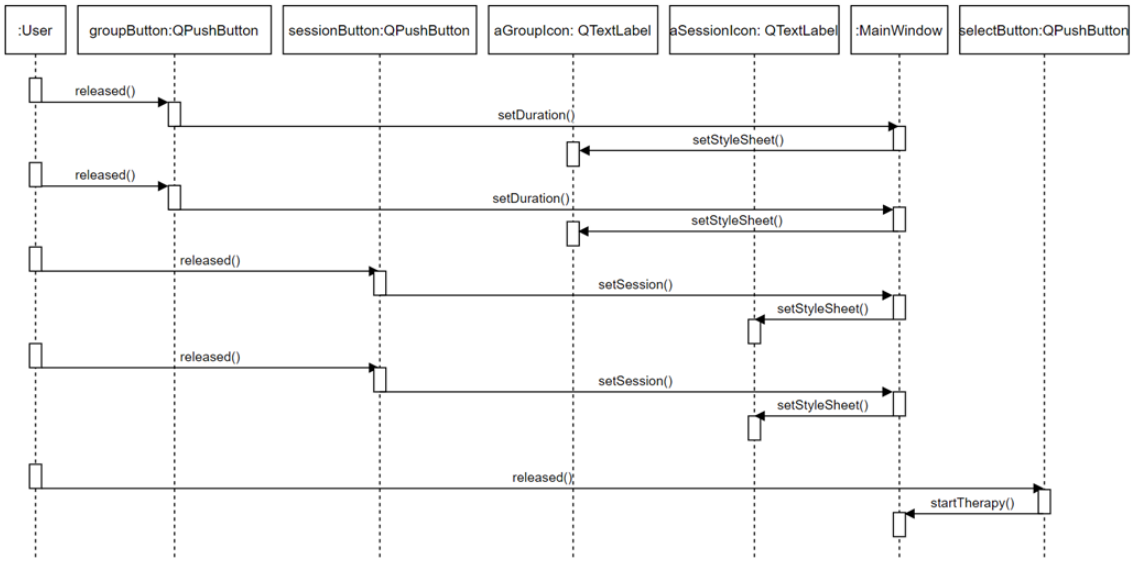
When the user starts a session, the connection test will be instantiated. The first thing that happens is that the ‘connectiontest’ class will check whether there is: No Connection, Okay Connection, or an Excellent Connection. If the connection is Okay or Excellent, the connection test proceeds. Based on the connection given by the ‘connectiontest’ class, the ‘MainWindow’ will change the colour of the QTextLabels (through setStyleSheet) to display the connection status on the CES device. The ‘connectiontest’ class will then notify the ‘MainWindow’ to blink the modelight every second. The ‘MainWindow’ will start a QTimer every second, where it will call the ‘blink_modelight’ function. Once the connection test finishes, the QTimer stops (stop blinking the mode light), the display on the UI is cleared, and all of the QTextLabels are changed back to its original background colour (setStyleSheet).



SELECTING A SESSION

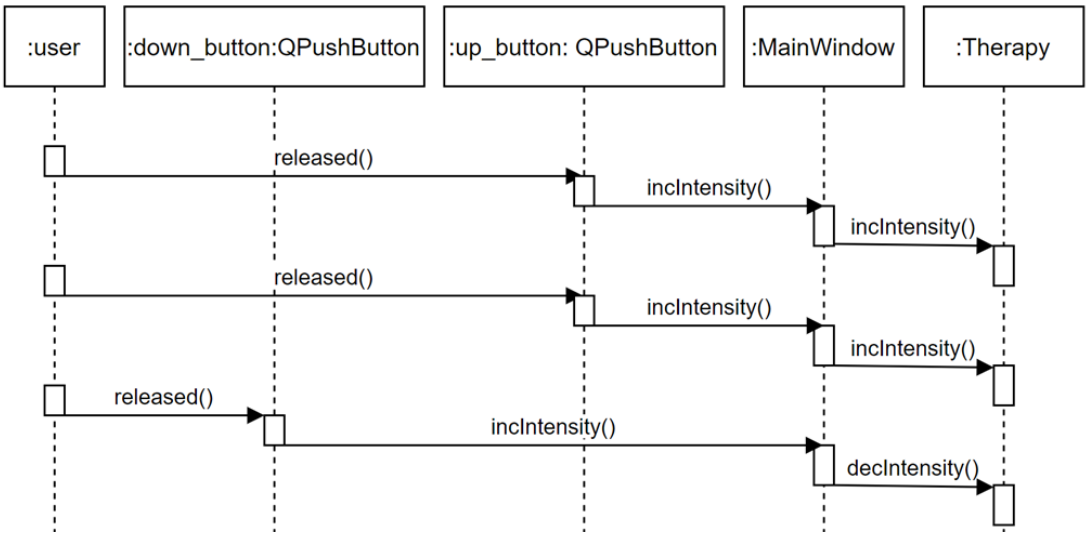
- When the device is turned on, the user can toggle between the three groups by clicking the groupButton on the CES Device (the QPushButton beside the 3 groups on the UI). When the button is released, it will trigger the ‘setDuration’ function to be called from within the ‘MainWindow’. The ‘MainWindow’ will change the stylesheet for the group icon lights accordingly (to display which group the cursor is currently on).

- Similarly, the user can toggle between the four different sessions (Alpha, Theta, Delta and 100HZ) by pressing and releasing the QPushButton beside the different sessions on the UI. The release of the QPushButton will trigger the ‘setSession’ function to be called from within the ‘MainWindow’. The ‘MainWindow’ will change the stylesheet for the session icon lights accordingly (to display which session the cursor is currently on).



ADJUSTING INTENSITY

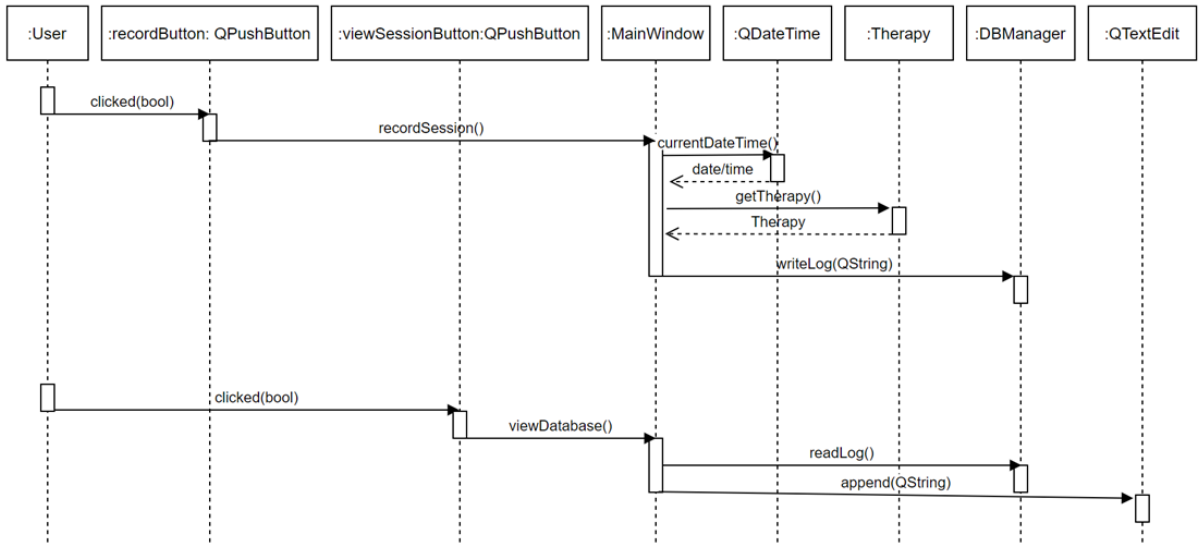
When a session has started, the user can press and release the up and down QPushButtons labeled ‘INT’ on the UI to change the intensity. When the ‘up’ button is released, it will trigger the ‘incIntensity’ function to be called from within the ‘MainWindow’. The ‘MainWindow’ will then call the ongoing therapy instance to increase the intensity. The same concept follows for when the user presses the ‘down’ button while a session is running.



RECORDING A SESSION/VIEWING RECORDED SESSIONS

- While a session is currently ongoing, the user can click the ‘Record Session’ QPushButton to record the ongoing session. When the button is clicked, it will call the ‘recordSession’ function from within the ‘MainWindow’. This function will get the current date and time from the ‘QDateTime’ QT class. It will also get the therapy information from the Therapy class. Then, it will create an organized QString to add to the database. Finally, it calls the ‘writeLog’ function from the ‘DBManager’ class to store the QString in a file (our database).

- To view all the recorded sessions, the user can click the ‘View Session Logs’ QPushButton. When clicked, it will trigger the ‘viewDatabase’ function to be called from the ‘MainWindow’. This function grabs the history of recorded sessions by calling the DBManager’s ‘readLog()’ function. It then displays the history of sessions to the QTextEdit by appending the information from readLog.



Final Message:

The above sequence diagrams relate to our use cases. We implemented our code strictly following our use cases. Now, we use the implementation (including proper function names) to create our sequence diagrams, which ultimately gives us a more accurate sequence diagram that relates to our use cases.

5. Design Patterns Used

There are two main patterns used in the system: Mediator, and Observer pattern. As a project built upon OO Design, and with multiple Classes with each their unique functionalities, we required a class (the mediator) that will represent the main machine and organize all the interactions between classes and the widgets. We chose that main class to be the MainWindow class. The MainWindow class will represent the main CES Device, and interact with all other classes each representing a specific feature, and each widgets on the Ui to organize the actions necessary, and represent the result to us via GUI. It was the most optimal strategy to use because we had a set of objects that needed to communicate in well-defined, complex ways, and there were behaviors that needed to be distributed between several classes without a lot of subclassing. As a result, we were able to abstract how objects cooperate, simplify object protocols, and centralize control of our device to one class.

The observer pattern is also strongly used in our project, as it is common for a project with GUI. The Ui class needs to constantly observe the subject (MainWindow) to catch any state changes via signals / slots and should update the GUI widgets. The observer pattern works well with the mediator pattern, since it limits the number of subjects the observer (Ui) needs to watch out for. All the state changes will be organized into the MainWindow class, and so the Ui can just look for signals from the MainWindow to be able to catch all the activities from the entire CES Device.

As more of a minor pattern, the singleton pattern was also used in the project. We had several classes that only needed to have a single copy of. For example, the MainWindow class represented the actual main brain of the CES Device, and so we only needed to have one instance of it, and it was the same for other components. Only one instance of Battery, DBManager, Therapy, and ConnectionTest had to exist at one time, since we were representing just one device. With the singleton pattern, we were able to control access to solve instances in a flexible way to make sure our project was properly structured.

6. Traceability Matrix

ID	Requirement	Related Use Case	Fulfilled By	Test	Description
1	The application interface contains buttons, display and connection test and battery elements	N/A	mainwindow.ui	Run the simulator and observe ui	Qt's user interface framework is used to replicate the physical CES device. An additional display and buttons are added to simulation to offer more actions in addition to the buttons found on the device.
2	Battery level depends on therapy intensity and time	N/A	mainwindow, battery, duration	Run the simulation and watch the battery level increase after every session starts	The battery level starts at 800 and will be depleted in proportion to the intensity level and duration of the treatment.
3	Treatment only occurs when connection test is passed	5	mainwindow, connectionTest	Leave all the connection test checkboxes unchecked, and treatment won't occur	The connection test must be passed in order for the treatment to run. The results of the connection test will be displayed. If an "ok" or an "excellent" connection occurs, the treatment will run. Otherwise, nothing happens.
4	Connection test is simulated using earlobes	5	mainwindow, connectionTest	Check the connection test earlobe checkboxes and the	The earlobe connection is simulated via the checkboxes in the gui. Connecting the earlobes results in an "ok"

				treatment will work	connection. The wetting of the earlobes may also be simulated to receive an “excellent” connection.
5	Device supports 4 frequencies	4	mainwindow, session	Use button beside session (frequency) images to select the frequencies and observe the logs	The device supports 4 different frequencies for the following sessions: alpha, delta, theta and 100Hz.
6	Device supports 4 groups (times)	4	mainwindow, duration	Use button beside time (group images to select the frequencies and observe the logs	The device supports 3 different groups including the following: 20 min, 45 min and user designated. The user designated time can be assigned from 1 to 180 minutes via a textbox.
7	User can record a treatment	7	mainwindow, DBManager	Press “Record Session” during session, then “View Session Logs” next time device is powered on	The user can record a treatment once it has started. During the softon and softoff, the session can’t be recorded.
8	User can view history of treatment	7	mainwindow, DBManager	Press “View Session Logs” next time device is powered on check for previous treatments	The user can view a history of their treatment once the device has been powered on. The device will store all the previously recorded treatments in persistent storage.

9	Device can be turned on and off	1, 2	mainwindow	Press power button and watch device lights turn on/off	The device can be turned on and off via the power button. The device can be turned off at any point, even during a treatment.
10	A record of treatment contains date, time and frequency (session type), group (time) and intensity	7	therapy, DBManager	Press “View Session Logs” next time device is powered on check for previous treatments	The session logs contain a list of previously recorded treatments with the following information: date, time, session and intensity.
11	Device stops working when battery level reaches 0	N/A	Mainwindow, battery	Attempt to power device when battery level is 0, without replacing battery, and see that device doesn’t turn on	As the device runs sessions, the battery level decreases. Once the battery level reaches 0, the device will power off and remain off until the battery is replaced via the gui.
12	Device turns off if no activity in 2 minutes	2	mainwindow	Wait 2 minutes without activity and see that device turns off	Once the device is powered on, a timer will commence. If there is no activity on the device after two minutes, all the lights will shut off and the buttons become non functional until the device is powered back on.
13	Battery level is displayed when device is turned on	3	mainwindow	Turn on device and see that the battery level is displayed via lights	The user can keep track of the battery level as its relative level is displayed via the lights on the device. The logs also display the precise battery

					level whenever a therapy is started.
14	All sessions end with a soft off	2	mainwindow	Wait for session to end, or press power button and watch softoff occur with lights slowly turning off	A soft off is the process where the therapy intensity slowly decreases until the intensity level reaches 0. The level lights will turn off one by one starting from light 8 to light 1.
15	Battery level displays 2 blinking bars when low	3	mainwindow,battery	Run a session multiple times until the battery level gets low.	Two blinking bars will occur when the battery is low. This indicates to the user that they should change the battery soon. However, the user can still run a therapy.
16	Battery level displays 1 blinking bar when critically low	3	mainwindow, battery	Run a session multiple times until the battery level gets critically low.	One blinking bar will occur when the battery is critically low. This indicates to the user that they should change the battery immediately. The user won't be able to run a therapy.
17	Session ends when battery is critically low	2	mainwindow, battery	Run a session multiple times until batter level gets critically low and observe the device power off	If the batter becomes critically low while therapy is running, the device will shut off. The user won't be able to use the device again until the batter has been replaced.

18	Therapy starts by pressing “select” button	4	mainwindow	Click select and observe lights and logs	Once a chosen group and session has been chosen, the user can start the therapy by pressing the “select” button. The button won’t start a therapy if the connection test has not passed, or the device is off.
19	Connection test occurs at the start of every therapy	5	therapy, connectionTest	Start a session and observe logs.	Once a therapy begins, the connection tests the connection between the device and the earlobes. The result of the connection test shows on the display.
20	The connection test results are reflected in the lights	5	mainwindow, connectionTest	Use the gui to simulate the connection test and observe the session's correspondi ng lights.	No connection blinks red, okay connection blinks yellow and excellent connection blinks green based on the results of the connection test.
21	Intensity blinks when intensity is adjusted	6	mainwindow, therapy	Start a session and observe blinking lights and logs when intensity is adjusted	After a therapy has begun, the intensity can be adjusted by pressing the up and down buttons. The corresponding intensity level light will blink to indicate the current intensity.