Final Deliverables Document
COMP3004 - Final Project
Josh Schiavone, Connor Lindblad, Maansinh Jadeja
Team 78

GitHub Repository: https://github.com/josh0xA/COMP3004-Final-Project

USE CASES

USE CASE 1: Turn on device

Primary Actor(s): Doctor

Stakeholders and Interests:
Patient - who is undergoing a test
Doctor - who is recording data presented from device

Pre-condition(s):
Device user interface is already designed and code successfully opens a simulation.

Success Guarantee(s): Run code on Qt and device user interface is presented. All buttons and lights are disabled before power buttons are pressed. Once the power button is pressed, the up button, down button, select buttons are enabled. User is able to successfully scroll through the menu presented on the home screen.

Main Success Scenario:

1. Power button is pressed
2. Up button is enabled
3. Down button is enabled
4. Select button is enabled
5. Session status lights are off but ready to run
6. Start button is disabled but ready to run
7. Stop button is disabled but ready to run
8. Pause button is disabled but ready to run

Extensions:

1a. Battery is dead
    1a1. Make sure battery is charged before pressing power button
2a. Up button is not responding to power button
    2a1. Make sure up button code allows it to be enabled and functions appropriately when power button is pressed
3a. Down button is not responding to power button

3a1. Make sure down button code allows it to be enabled and functions appropriately when power button is pressed

4a. Select button is not responding to power button

2a1. Make sure select button code allows it to be enabled and functions appropriately when power button is pressed

5a. Session status lights are not off

5a1. Make sure session status lights code allows it to only function during session

6a. Start button is not responding to power button

6a1. Make sure start button code allows it to only function during session

7a. Stop button is not responding to power button

7a1. Make sure stop button code allows it to only function during session

8a. Pause button is not responding to power button

8a1. Make sure start button code allows it to only function during session

USE CASE 2: Start new session

Primary Actor(s): Doctor

Stakeholders and Interests:
Patient - who is undergoing a test
Doctor - who is recording data presented from device

Pre-condition(s):
Device user interface is already designed and code successfully opens a simulation and the menu screen is functioning appropriately.

Success Guarantee(s): Run code on Qt and device user interface is presented. All buttons and lights are disabled before power buttons are pressed. Once the power button is pressed, the up button, down button, select buttons are enabled. User is able to successfully scroll through the menu presented on the home screen. User selects new session, start, pause and stop button are enabled. Once start button is pressed green light turns on for treatment signal.

Main Success Scenario:

1. Power button is pressed
2. New session is selected
3. New session screen is presented
4. Start button is enabled
5. Stop button is enabled
6. Pause button is enabled
7. Once start button is pressed timer and progress tracker begins
9. Treatment signal (green light) turns on.

10. If the stop button is pressed then the contact lost (red light) light turns on and timer and progress is stopped.
11. If the pause button is pressed then the timer and progress is stopped.
12. Current date gets added to session log list

Extensions:

1a. Battery is dead
    1a1. Make sure battery is charged before pressing power button
2a. New session is not allowing to be selected
    2a1. Make sure the select button allows the new session option to be selected.
3a. New session screen is not presented
    3a1. Make sure code presents a new session screen which shows a timer and progress when the start button is pressed.
4a. Start button is not enabled
    4a1. Make sure code enables the start button when the new session option is selected.
5a. Stop button is not enabled
    5a1. Make sure code enables the stop button when the new session option is selected.
6a. Pause button is not enabled
    6a1. Make sure code enables the pause button when the new session option is selected.
7a. Timer and progress tracker do not start when the start button is pressed.
    7a1. Make sure code allows timer and progress tracker to begin once the start button is pressed.
9a. Treatment signal light does not turn on
    9a1. Make sure code switches on treatment signal green light once the start button is pressed.
10a. Contact lost light does not turn on
    10a1. Make sure code switches on the contact lost red light when the stop button is pressed and stops the timer and progress tracker.
11a. Pause button does not pause the session.
    11a1. Make sure code stops the timer and progress tracker when the pause button is pressed.
12a. Current date does not get added to session log list
    12a1. Make sure code saves the current date to the session log list.

USE CASE 3: View session log

Primary Actor(s): Doctor

Stakeholders and Interests:

Patient - who is undergoing a test
Doctor - who is recording data presented from device

Pre-condition(s):

Device user interface is already designed and code successfully opens a simulation and the menu screen is functioning appropriately.

Success Guarantee(s): Run code on Qt and device user interface is presented. All buttons and lights are disabled before power buttons are pressed. Once the power button is pressed, the up button, down button, select buttons are enabled. User is able to successfully scroll through the menu presented on the home screen. User selects session log, and a list of previous sessions is presented.

Main Success Scenario:

1. Power button is pressed
2. Session log is selected
3. List of previous session dates is presented.

Extensions:

1a. Battery is dead
    1a1. Make sure battery is charged before pressing power button
2a. Session log is not allowing to be selected
    2a1. Make sure the select button allows the session log option to be selected.
3a. Previous sessions are not being saved to session log
    3a1. Make sure code saves dates to a list of session logs of each session and is presented in the session log menu.

USE CASE 4: Edit time and date

Primary Actor(s): Doctor

Stakeholders and Interests:
Patient - who is undergoing a test
Doctor - who is recording data presented from device

Pre-condition(s):
Device user interface is already designed and code successfully opens a simulation and the menu screen is functioning appropriately.

Success Guarantee(s): Run code on Qt and device user interface is presented. All buttons and lights are disabled before power buttons are pressed. Once the power button is pressed, the up button, down button, select buttons are enabled. User is able to successfully scroll through the menu presented on the home screen. User selects time and date and is able to change the current date and time for the session.

Main Success Scenario:

1. Power button is pressed
2. Time and date is selected
3. Time and date screen is presented with option to change details for the user
4. User is able to change time and date

Extensions:
1a. Battery is dead
    1a1. Make sure battery is charged before pressing power button
2a. Time and date is not allowing to be selected
    2a1. Make sure the select button allows the time and date option to be selected.
3a. Time and date screen is not presented
    3a1. Make sure code presents screen for time and date where the user is able to manually change the current time and date.
4a. Time and date is not being changed
    4a1. Make sure code saves the selected time and date and uses those details for when user starts a new session.

USE CASE 5: Battery Depletion

Primary Actor(s): Doctor

Stakeholders and Interests:
Patient**:** Ensuring the device operates throughout the session without interruptions due to power loss.
Doctor**:** Ensuring that the session can be completed and data is not lost due to battery issues.

Pre-condition(s):
Device user interface is designed and operating correctly.
The device is powered on and operational.

Success Guarantee(s):
The device monitors battery levels and warns the user if the battery is low.
The device allows for battery recharge when levels are low.

Main Success Scenario:
Battery level is monitored throughout the device operation.
If battery falls below 20%, a warning is displayed to the user (battery bar goes red).
User can initiate a battery charge when the battery dies.

USE CASE 6: Contact Loss

Primary Actor(s)**:** Doctor

Stakeholders and Interests**:**

Patient: Concerned about the uninterrupted operation of the device and accurate recording of session data.

Doctor: Needs to maintain consistent contact with the device to ensure the quality and continuity of the data collected.

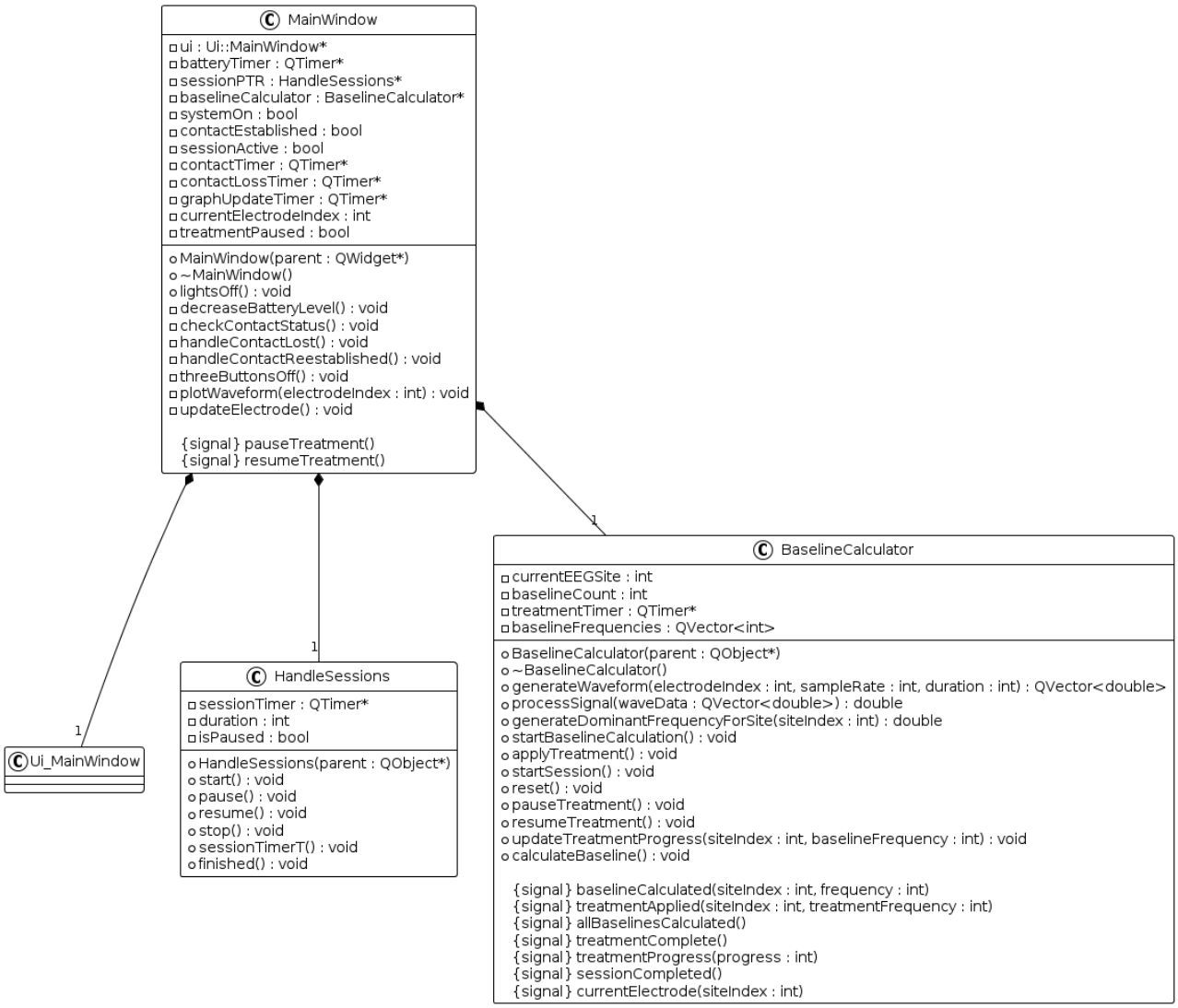
Pre-condition(s)**:**
Device user interface is designed and operational.
A session is currently in progress.

Success Guarantee(s):
The device continuously monitors the contact status.
The user is warned if contact with the device is lost.
The session can be paused and resumed if contact is reestablished within a certain timeframe.

Main Success Scenario**:**
Contact status is checked regularly during a session.
If contact is lost, the session is paused, and a warning is displayed.
If contact is not reestablished within a designated timeframe, the session is stopped, and the device shuts down.
User can manually attempt to reestablish contact if the automated process fails.

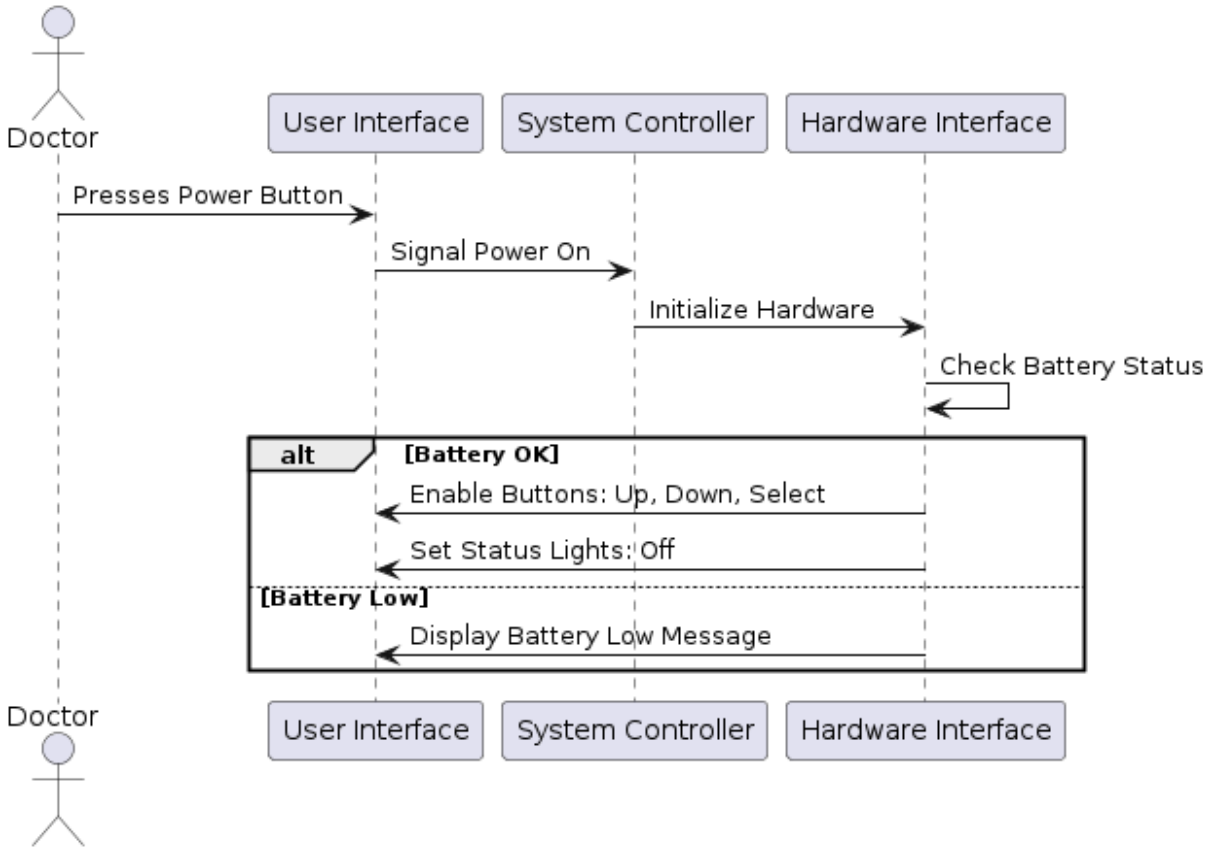
Extensions:
1a. Contact is lost, and automatic reconnection attempts fail:
1a1. The doctor can attempt to manually reestablish contact through the UI.
2a. The session cannot continue due to prolonged loss of contact:
2a1. The device saves the session data up to the point of contact loss and powers down to prevent further data corruption.

UML Class Diagram

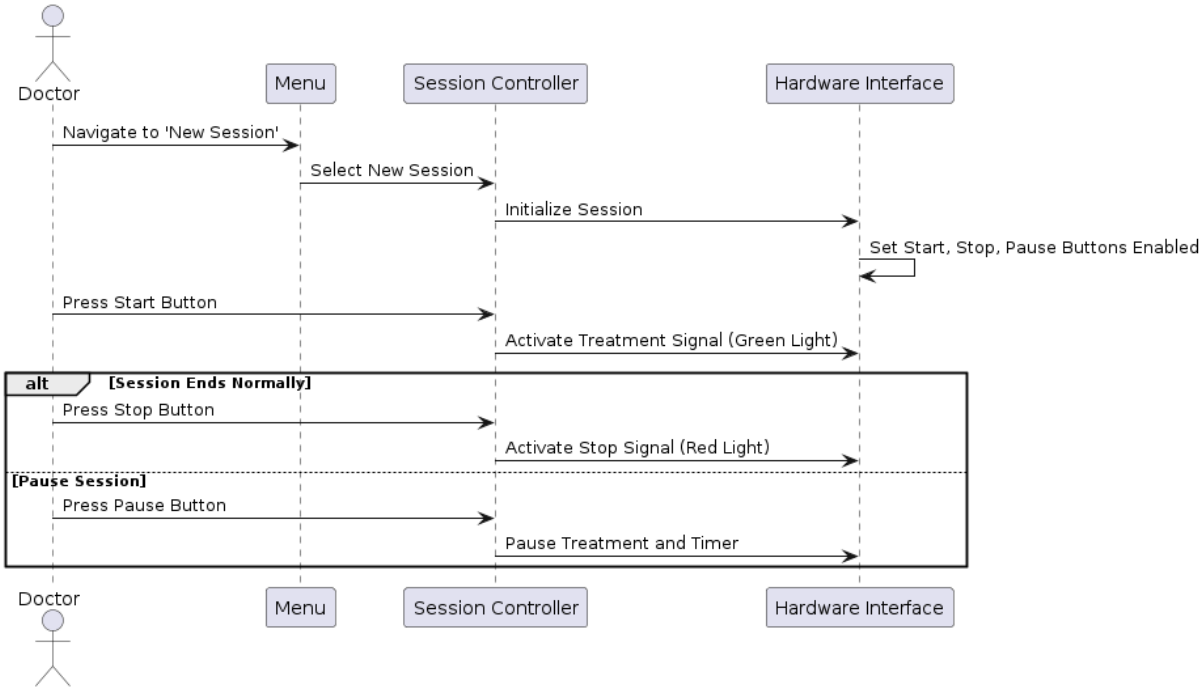


SEQUENCE DIAGRAMS

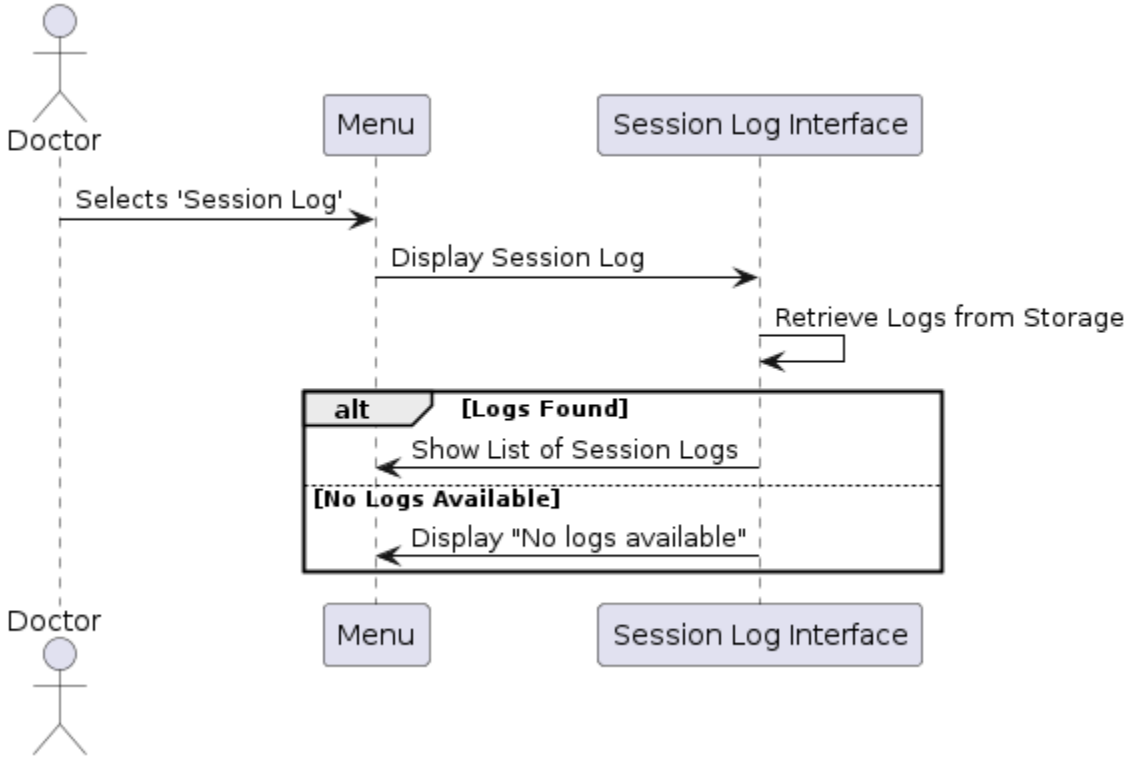
A) Device Activation



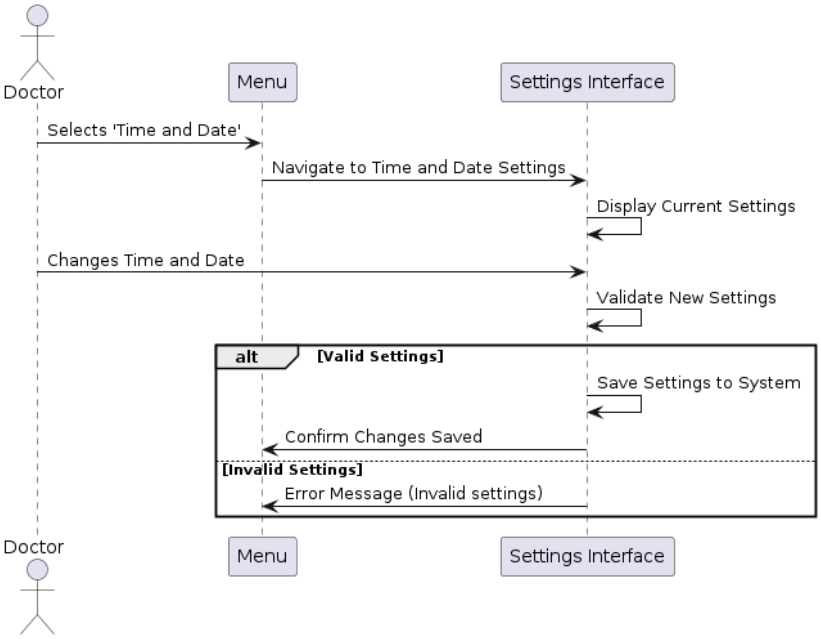
B) Session Start



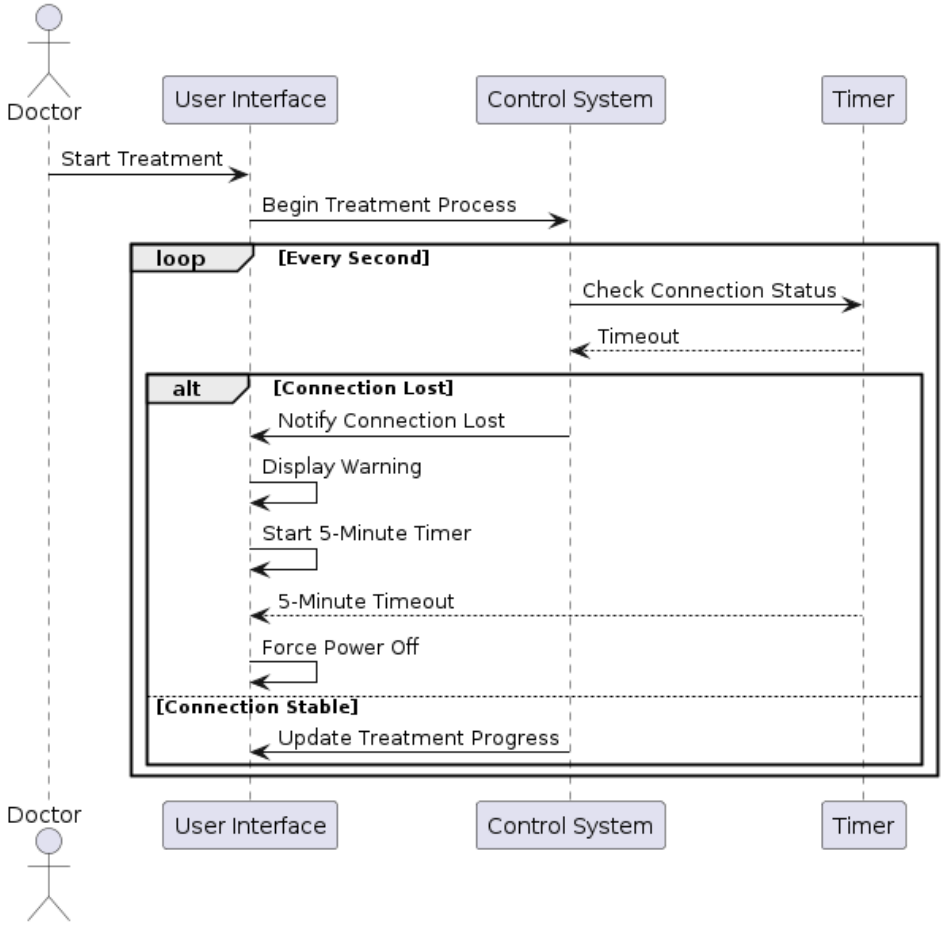
C) Session Log



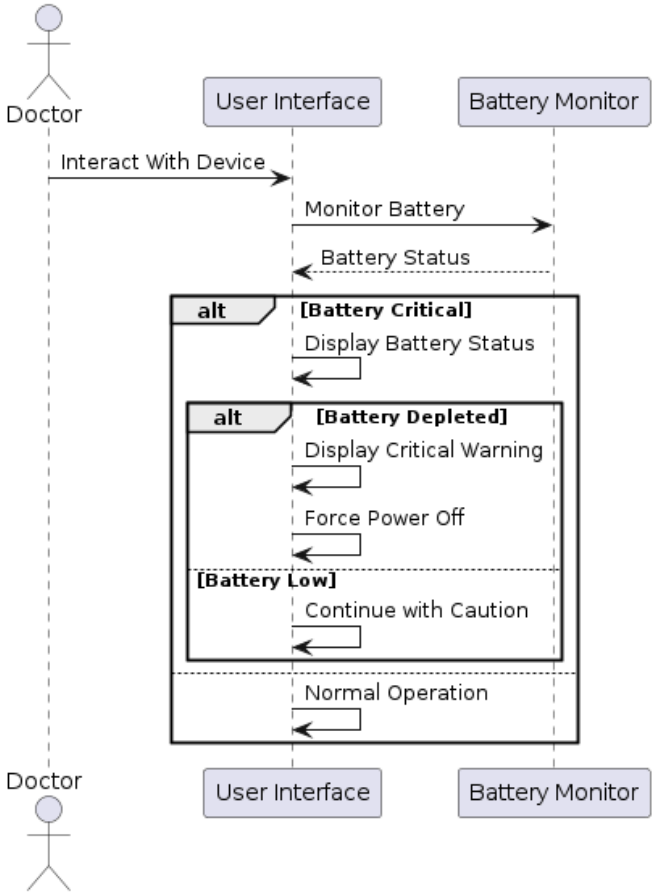
D) Time and Date entry



C) [Safety] Contact Loss

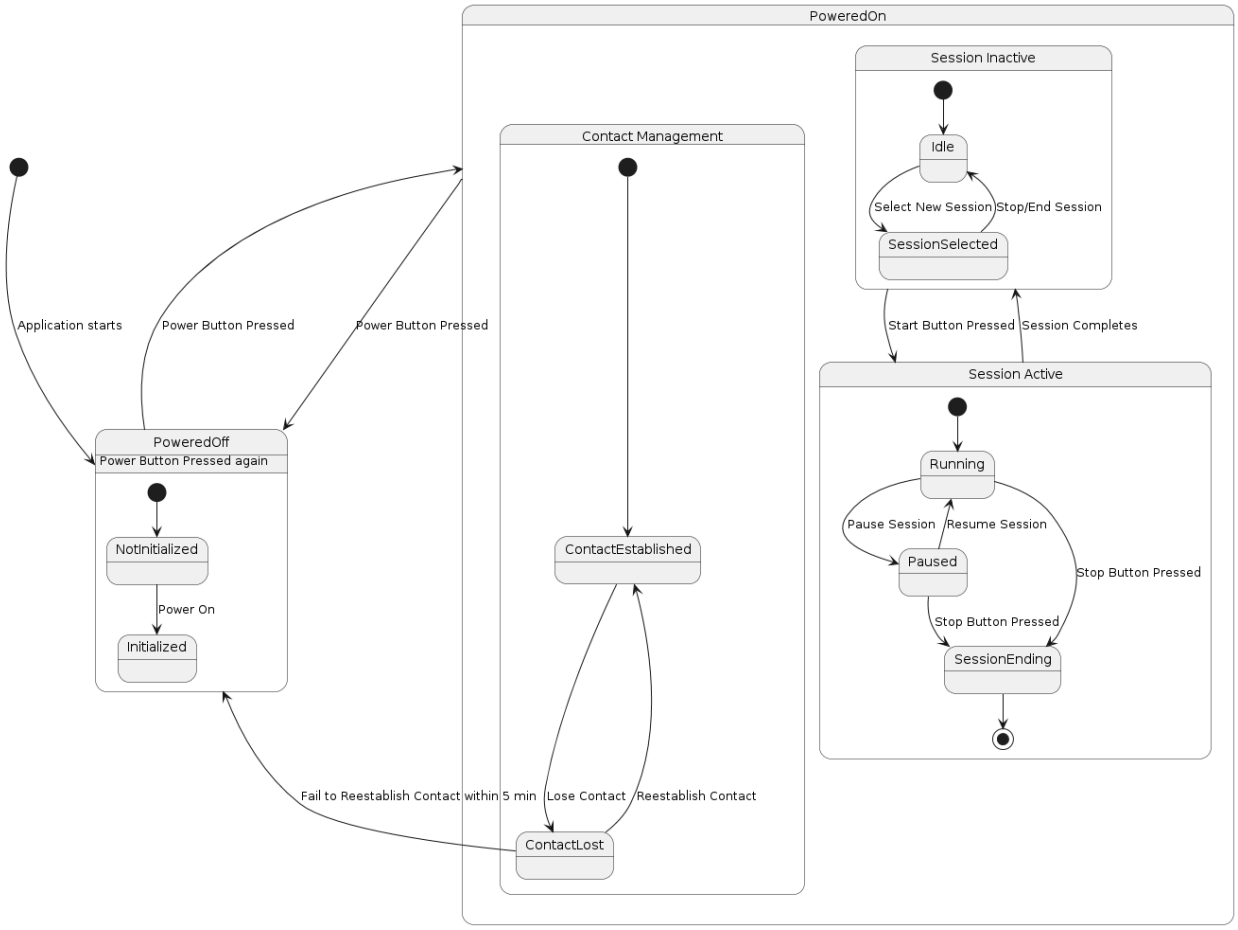


D) [Safety] Battery Low

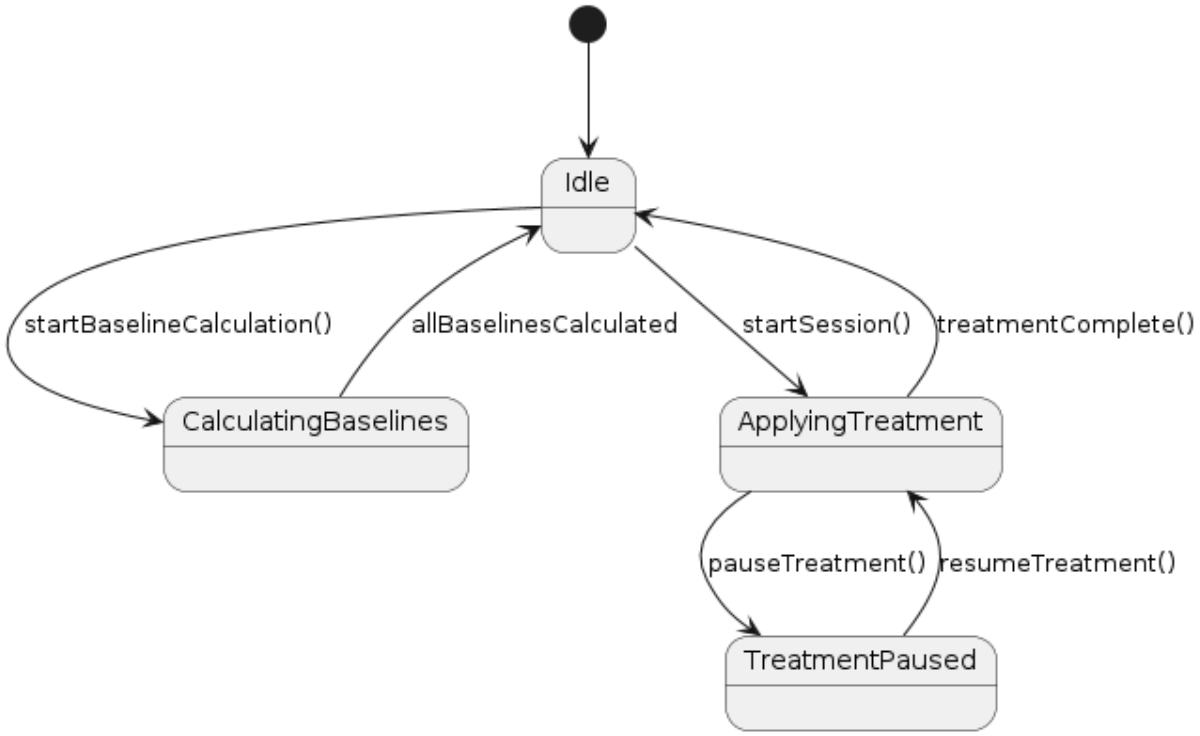


State Diagrams:

UML State Diagram for MainWindow Control Entity



UML State Diagram for BaselineCalculator Control Entity



Traceability Matrix

ID	Description	Use Case	Design	Fulfilled By	Test
1	Powering on device	Turn on device	A signal is sent alerting the system that the power button has been pressed. The menu, battery percentage, navigation arrows and select will now displayed and interactable.	-Mainwindow	Run the QT project, the UI will display the device and the power button, which can be clicked to observe this functionality.

2	Start a new session	Start new session	When a new session is started, a signal is sent to the baseline calculator which will run and display the treatment session along with the current progress. Upon completion handle sessions will run and emit a done signal to the main window, which saves the session.	-Mainwindow -Baselinecalculator -Handlesessions	Select NEW SESSION, once selected a new display will be shown with a start button which can be pressed to view this.
3	Pause a session	Start new session	When a session is running and the pause button is pressed, a signal is sent to handle sessions which will run necessary checks and then emits a paused signal, which will pause the timer in the baseline calculator.	-Mainwindow -Handlesessions -Baselinecalculator	While a session is running there will be a pause button enabled you can click to view this.
4	Resume a session	Start new session	When a session has been paused the progress will not be moving. When the pause button is clicked again, a signal will be sent to handle sessions which runs checks and sends another signal to the baseline calculator to resume.	-Mainwindow -Handlesessions -Baselinecalculator	While a session is paused click the pause button again to view this.
5	Stop a session	Start new session	When a session is running and the stop button is pressed, a signal is sent to handle sessions which will run necessary checks and then emits a quit signal, which will stop the timer in the baseline calculator and cancel all progress so that a new session must be started from scratch.	-Mainwindow -Handlesessions -Baselinecalculator	While a session is running there will be a stop button enabled you can click to view this.
6	Set date	Edit time and date	After the TIME AND DATE option has been clicked, a display is shown which includes dates ordered MM/DD/YY. The default time can be clicked on and changed to the user's	-Mainwindow	While in the menu navigate to TIME AND DATE. There is a date and time which can be clicked and

			desired time. This time will be automatically saved and displayed after a session is completed.		changed.
7	View sessions	View session log	While on the main menu UI, the sessions that have been completed have been saved into a display which can be viewed in SESSION LOG.	-Mainwindow	While in the menu navigate and click on SESSION LOG, and any sessions will appear.
8	Power off	NA	When the device is on and the power button is pressed it will send a signal to turn off the UI but still save all current sessions.	-Mainwindow	While the UI is on, click the power button to view this.
9	Battery	Battery Depletion	The system monitors the battery level continuously. When the level falls below 20%, it sends a warning signal by turning the battery bar red. .	-Mainwindow	Power on the UI while the battery timer is active and depletes every 1 second, when it's at 20%, ensure the battery progress bar emits red.
10	Contact Loss	Contact Loss	The system periodically checks for contact with the EEG device. If contact is lost, a signal is sent to the MainWindow, which then handles the event by stopping the session, notifying the user, and attempting to reestablish contact.	-Mainwindow	Trigger a 'loss of contact' event during treatment and verify that the session stops, a message is displayed, and an attempt to reconnect is made. If contact is not reestablished within a specified timeout, the device should turn off.

Design Decisions:

CLASSES

BaselineCalculator.h

The `BaselineCalculator` class provides functionality for generating baseline EEG frequencies and applying treatments to multiple electrode sites. It manages the calculation of baseline frequencies for each site, which are then emitted through signals. Additionally, it handles the application of treatments by incrementing the baseline frequency and updating treatment progress accordingly. The class utilizes timers to control treatment duration and ensures responsiveness by processing events to prevent UI blocking during treatment application. Overall, `BaselineCalculator` facilitates the initialization and execution of EEG sessions, enabling the application to interact with EEG devices effectively.

HandleSessions.h

The `HandleSessions` class is responsible for managing session timing and progression. It utilizes a QTimer to track session duration, emitting signals to notify the application of session updates, pauses, resumptions, completions, and quits. The class offers methods to start, pause, resume, and stop sessions, as well as a signal to indicate when the session is finished. Additionally, it provides a boolean function to check if the session is paused. The session timer increments until it reaches a predefined duration, at which point it emits the 'finished' signal and stops. Overall, `HandleSessions` facilitates the control and monitoring of session lifecycles within the application.

MainWindow.h

The `MainWindow` class serves as the primary interface for the application, managing user interactions, session controls, and UI updates. It inherits from `QMainWindow` and contains various member variables, including pointers to timers, instances of other classes like `HandleSessions` and `BaselineCalculator`, and flags to track system states and user interactions.

The class provides slots for handling button clicks, such as starting, pausing, resuming, and stopping sessions. It also manages battery level updates and contact status monitoring. Signals and slots are utilized to communicate between the UI elements and the backend logic.

The `MainWindow` class coordinates with instances of `HandleSessions` and `BaselineCalculator` to manage session timing, baseline calculations, and treatment application. It updates the UI elements based on session progress and user interactions, such as button clicks and contact status changes.

Overall, the `MainWindow` class acts as the central hub for the application, orchestrating the flow of information and interactions between the user interface and the underlying functionality provided by other classes.

QCustomPlot:

We utilize `QCustomPlot` as our external graphing library, harnessing its robust capabilities to integrate dynamic and visually appealing graphs and charts seamlessly into our applications. With `QCustomPlot`, we benefit from a versatile toolkit that enables us to create an array of customizable data visualizations, including line plots, scatter plots, histograms, and more. Leveraging this library, we have the flexibility to tailor the appearance and behavior of our graphs to meet the specific requirements and aesthetic preferences of our applications and users. Additionally, `QCustomPlot` offers interactive features such as zooming, panning, and data point selection, enhancing user engagement and facilitating in-depth data exploration and analysis. As a cornerstone of our graphing functionality, `QCustomPlot` empowers us to deliver intuitive and informative data visualization experiences to our users.