# Comp 3004: Neureset – Direct Neurofeedback EEG Device

Alvina Han, Aekus Trehan, James Yap, Lei Wu, Matthew Seto

## Use Cases

**Use Case 1: Turn Device On**
Primary Actor: Device User
Scope: Neureset Device
Level: User Goal
Stakeholders and Interests:
Neureset User - wants to utilize Neureset
Precondition: User has a Neureset Device.
Minimal Guarantees: The device will not turn on if the battery is not charged.
Success Guarantee: The device has been turned on and is waiting on the main menu.
Main Success Scenario:

1. User presses the power button.
2. The device turns on.

Extensions:
2a. The device does not turn on.

- The user tries to charge the battery and if it still doesn't turn on, the device is faulty and the user contacts customer support.

**Use Case 2: Turn Device Off**
Primary Actor: Device User
Scope: Neureset Device
Level: User Goal
Stakeholders and Interests:
Neureset User - wants to utilize Neureset
Precondition: User has a Neureset Device that is turned on.
Minimum Guarantees: Nothing happens.
Success Guarantee: The device has been turned off and is no longer consuming power.
Main Success Scenario:

1. User pressed the power button.
2. The device turns off.

**Use Case 3: User Selects a Program**
Primary Actor: Device User
Scope: Neureset Device

Level: User Goal
Stakeholders and Interests:
Neureset User - wants to utilize Neureset
Precondition: User has a Neureset Device that is turned on and is on the main menu.
Minimum Guarantees: The device will turn off if it runs out of battery.
Success Guarantee:
The user has initiated one of the three programs of either:

- Starting a treatment
- Viewing past treatments
- Change date and time.

Main Success Scenario:

1. User navigates through the menu and decides on what they want to do.
2. User selects their desired program by pressing a button.
    a. User has selected to start a new session. Proceed to *Use Case 4*.
    b. User has selected to View Past Session. Proceed to *Use Case 5/6*.
    c. User has selected to Change Date and Time. Proceed to *Use Case 7*.

**Use Case 4: User Treatment**
Primary Actor: Device User
Scope: Neureset Device
Level: User Goal
Stakeholders and Interests:
Neureset User - wants to utilize Neureset
Precondition:

- User has selected to start a treatment
- EEG electrodes contact is established (indicated by a blue light)

Minimum Guarantees: The device turns off if it runs out of battery or if electrodes are disconnected for too long.
Success Guarantee: The device has completed a treatment and logs the treatment into logged session/past treatments.
Main Success Scenario:

1. User contacts the Neureset and EEG electrodes are established.
2. User starts a new session.
3. Neureset device starts a timer.
4. Device displays session progress with an approximate time remaining and a percentage progress bar.
5. Electrodes read a signal from one of the 21(7) sites on the headset.
6. Device establishes a baseline average frequency.
7. Devices add an offset frequency of 5hz to the baseline frequency.
8. Device recalculates baseline frequency after the addition of the offset.
9. Device repeats step 5 until total offset frequency added is 20hz.

10. Repeat step 4 for each of the Electrodes.
11. Session ends when the timer reaches zero.
12. Device notifies the user that the treatment has been completed.
13. Device logs the previous treatment to the database (Past Treatments).

Extensions:

1a. Proceed to Use Case 9: Connection Lost Between Electrodes and the Device.

2ai. Users can voluntarily pause the session by pressing the pause button.

2aii. If contact is not reestablished within 5 minutes of pausing, the device automatically terminates the session and turns off.

2b. If the session is interrupted due to external factors (e.g., power outage), the device saves session progress and prompts the user to resume or start a new session upon power restoration.

**Use Case 5: User View Past Treatments On Neureset**

Primary Actor: Device User

Precondition:

- The device treatment has just concluded
- User selects View Past Treatments from the menu

Scope: Neureset Device

Level: User Goal

Stakeholders and Interests:

Neureset User - wants to utilize Neureset

Minimum Guarantee: A blank page is displayed when no previous treatments have been completed.

Success Guarantee: The device will display all the previous treatments completed.

Main Success Scenario:

1. Device displays the past treatments with timestamps (date and time).
2. User navigates the past treatments using a scroll.
3. When the user is done, they can press a different tab.

**Use Case 6: User View Past Treatments on PC**

Primary Actor: Device User

Precondition:

- The device treatment has just concluded
- User selects View Past Treatments from the menu

Scope: PC Device

Level: User Goal

Stakeholders and Interests:

PC User - wants to utilize PC to see baselines of past treatments

Minimum Guarantee: A blank page is displayed when no previous treatments have been completed.

Success Guarantee: The device will display all of the previous treatments completed.

Main Success Scenario:

1. Device displays the past treatments with timestamps (date and time).

2. User ensures that the Neureset is plugged into the PC.
3. User presses the sync changes button.
4. The Neureset will sync with the computer and upload its previous sessions with baselines.
5. User can now view past treatments with the timestamp and the starting and end baselines.

**Use Case 7: User Changes Time and Date**

Primary Actor: Device User
Precondition: User selects Set Time and Date from menu
Scope: Neureset Device
Level: User Goal
Stakeholders and Interests:
Neureset User - wants to utilize Neureset
Minimum Guarantee: The time and date remain unchanged on the device.
Success Guarantee: The time and date are changed on the device.
Main Success Scenario:

1. Device displays a time/date to the user.
2. The user can input a new time.
3. The device updates the display at the new updated time.
4. The new time should be reflected within new session logs.

**Use Case 8: Battery Low Response of the Device**

Primary Actor: Device User
Precondition: The device is low on battery
Scope: Neureset Device
Level: User Goal
Stakeholders and Interests:
Neureset User - wants to utilize Neureset
Minimum Guarantee: A blank page is displayed when no previous treatments have been completed.
Success Guarantee: The device will display all of the previous treatments completed.
Main Success Scenario:

1. Device battery is low.
2. Device battery is updated to showcase low battery.
3. Device displays text and audio to tell the user to charge the device.

**Use Case 9: Connection Lost Between Electrodes and the Device**

Primary Actor: Device User
Precondition:

- The device is performing a treatment
- Connection is lost between Electrodes and the Device

Scope: Neureset Device
Level: User Goal

Stakeholders and Interests:
Neureset User - wants to utilize Neureset
Minimum Guarantee: The device turns off and the session is terminated.
Success Guarantee: The device will continue performing the treatment.
Main Success Scenario:

1. Device loses contact with EEG electrodes.
2. The device flashes a red light.
3. The device pauses the session.
4. The display displays an error message and prompts the user to check electrode connections.
5. The device starts beeping until contact is reestablished.

Extension:
4a. If contact is not reestablished within 5 minutes, the device automatically turns off, and the session is terminated.

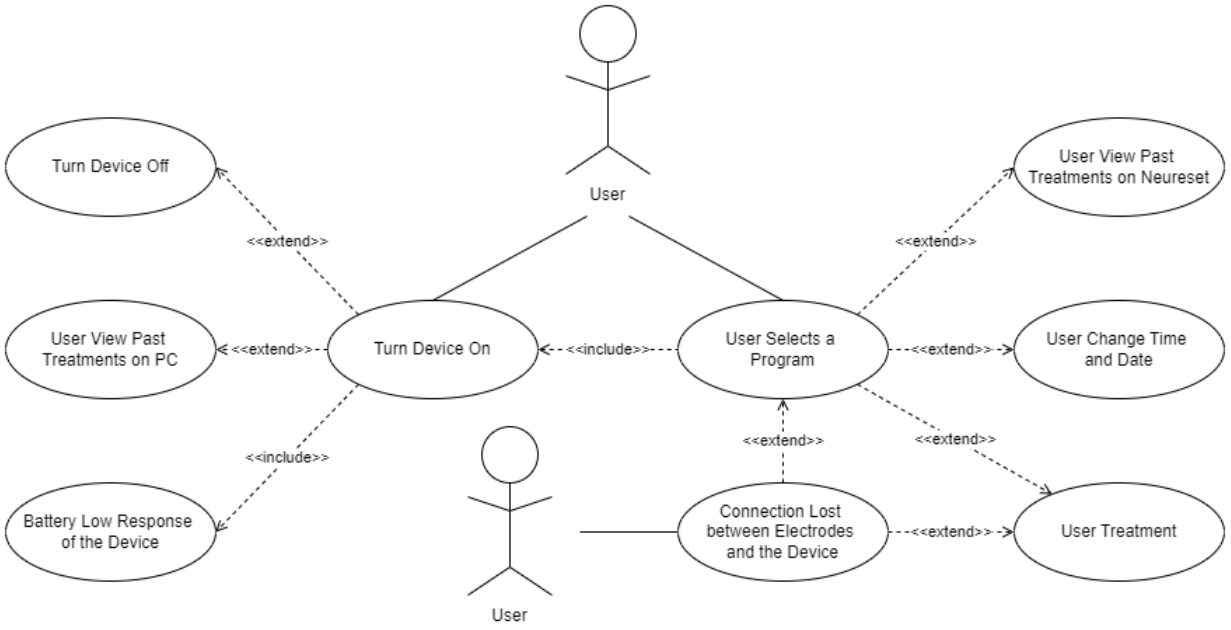


*Figure 1: Use Case Diagram*

It is mandatory for the user to turn the device on for executing any of the cases.

# Design Documentation

All diagrams are in their respective diagram folders under “Diagrams+Report” folder for reference (in GitHub).

- UML Class Diagram
- Sequence Diagram
- State Diagram
- (Use Case Diagram)

# UML Class Diagram

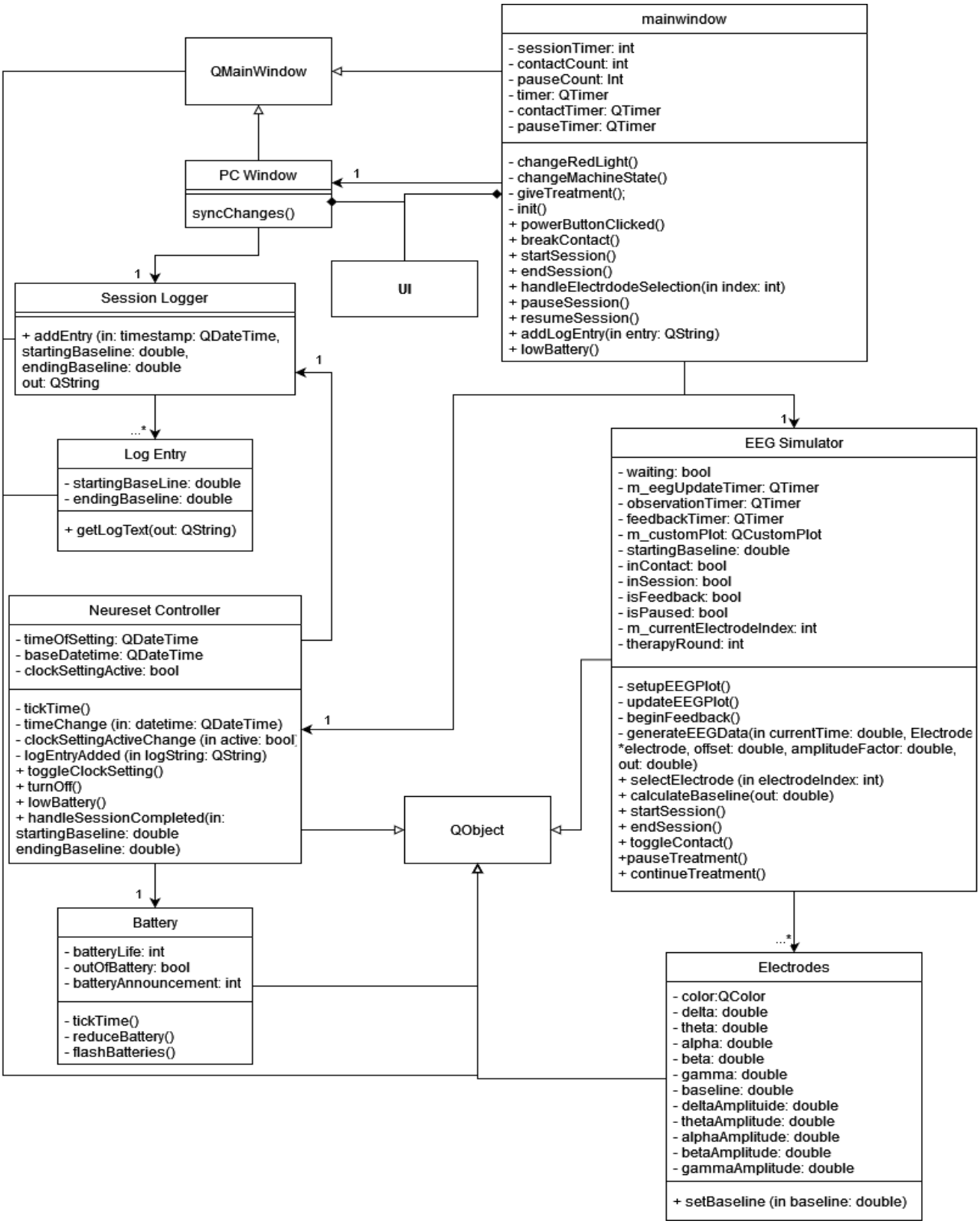


*Figure 2: UML Class Diagram*

The two main entities are Neureset Controller and EEG Simulator. The EEG Simulator is focused on the EEG waves and the offsets during the main scenario, user treatment. The Neureset Controller is a controller that controls the small entities that make up the device. Not a specific design is implemented but we have separated the functionalities into two: one handling the treatment and one that handles everything else.

## Sequence Diagram

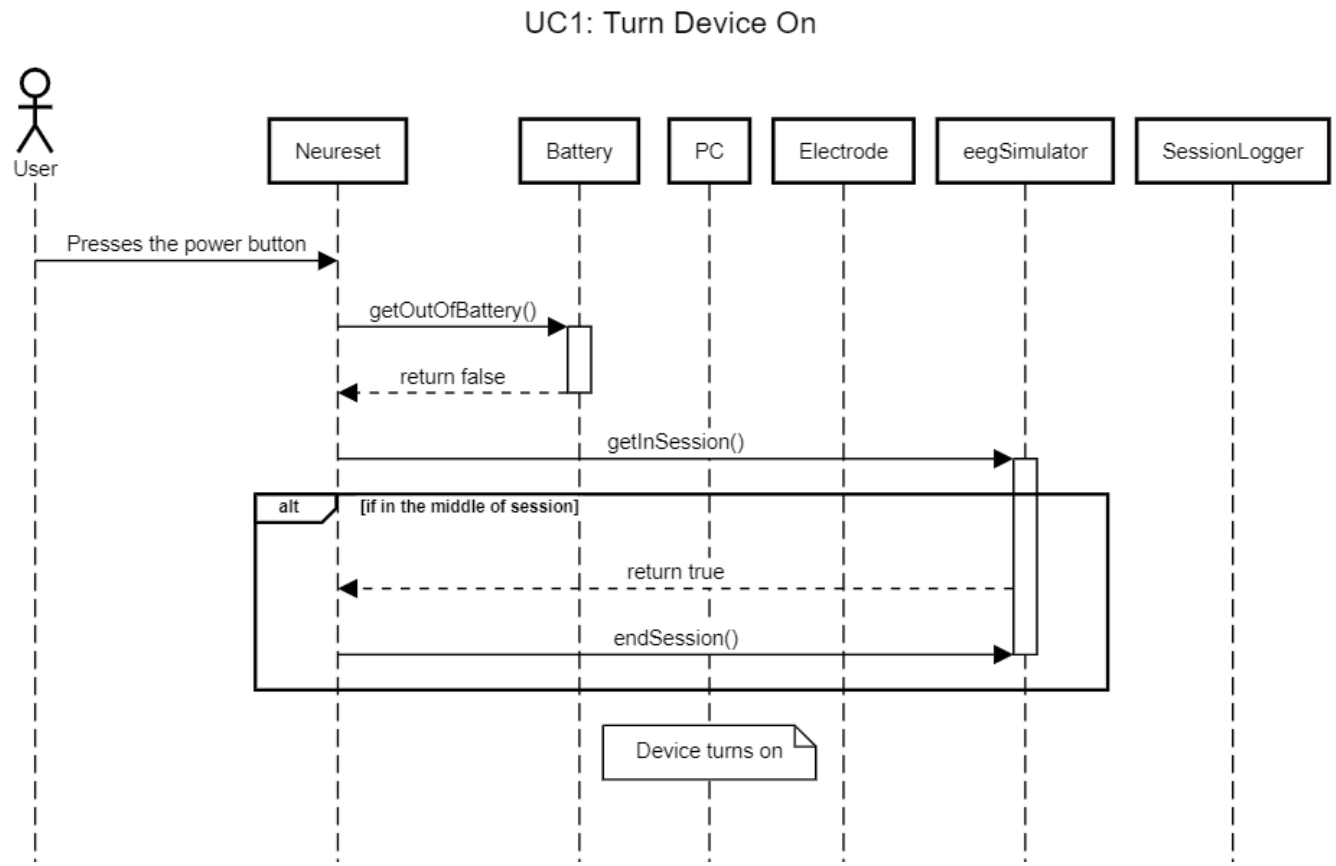


Figure 3 is success scenario of Use Case 1. It is a success scenario when an actor, user, presses the power button and once the device checks that it is not out of battery, it will turn on. One thing to note here is that it is assumed that the device can be turned off in middle of the session so once the device is turned on again, the session will be terminated and need to be restarted.

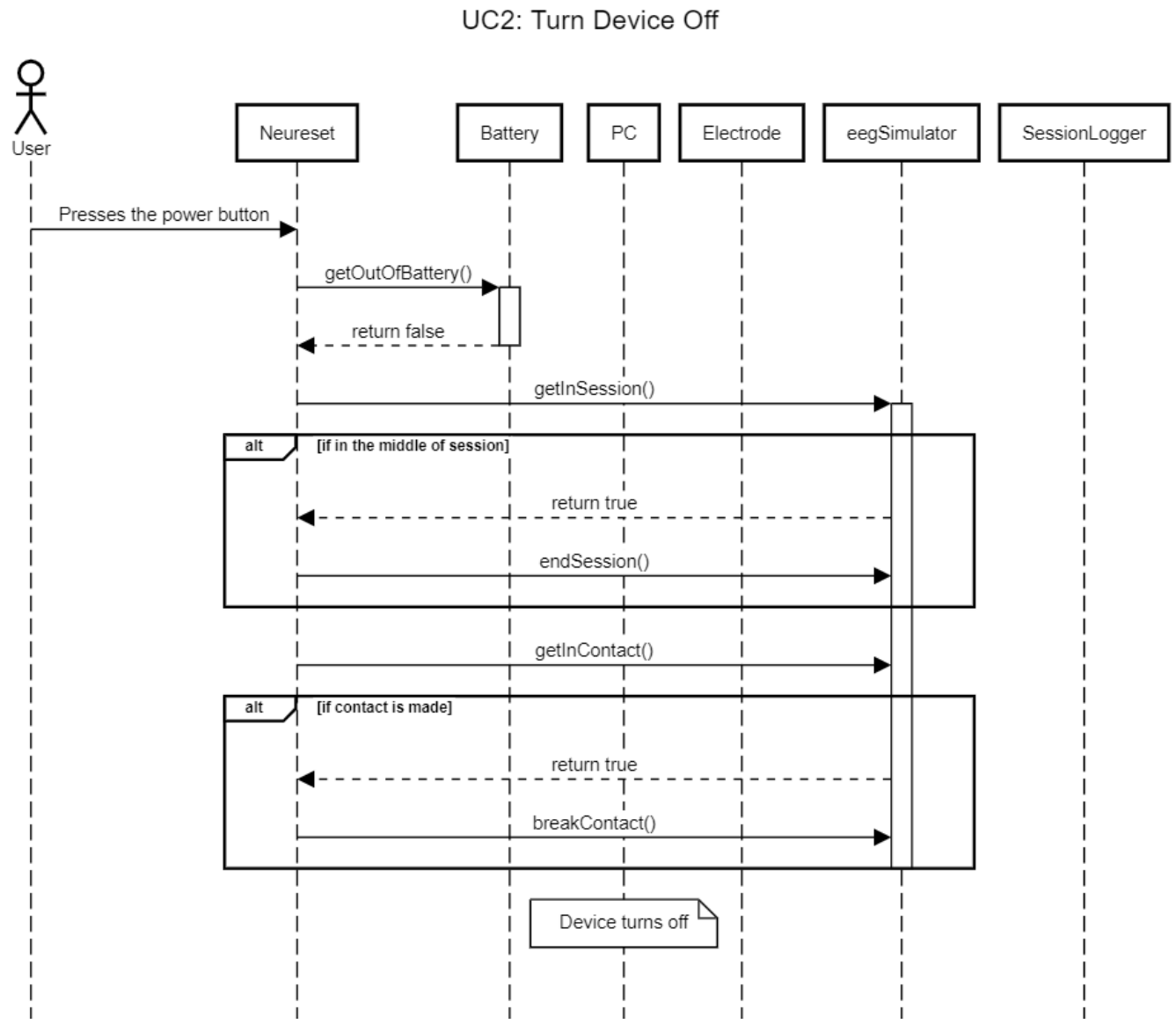


Figure 4 is success scenario of Use Case 2. It is a success scenario when an actor, user, presses the power button and once the device checks that it is not out of battery, it will turn off. One thing to note here is that it is assumed that the device can be turned off in middle of the session and if it does, it will end the session. Also, if contact is made at the point of turn off, the device will break contact with the user.

UC4: User Treatment

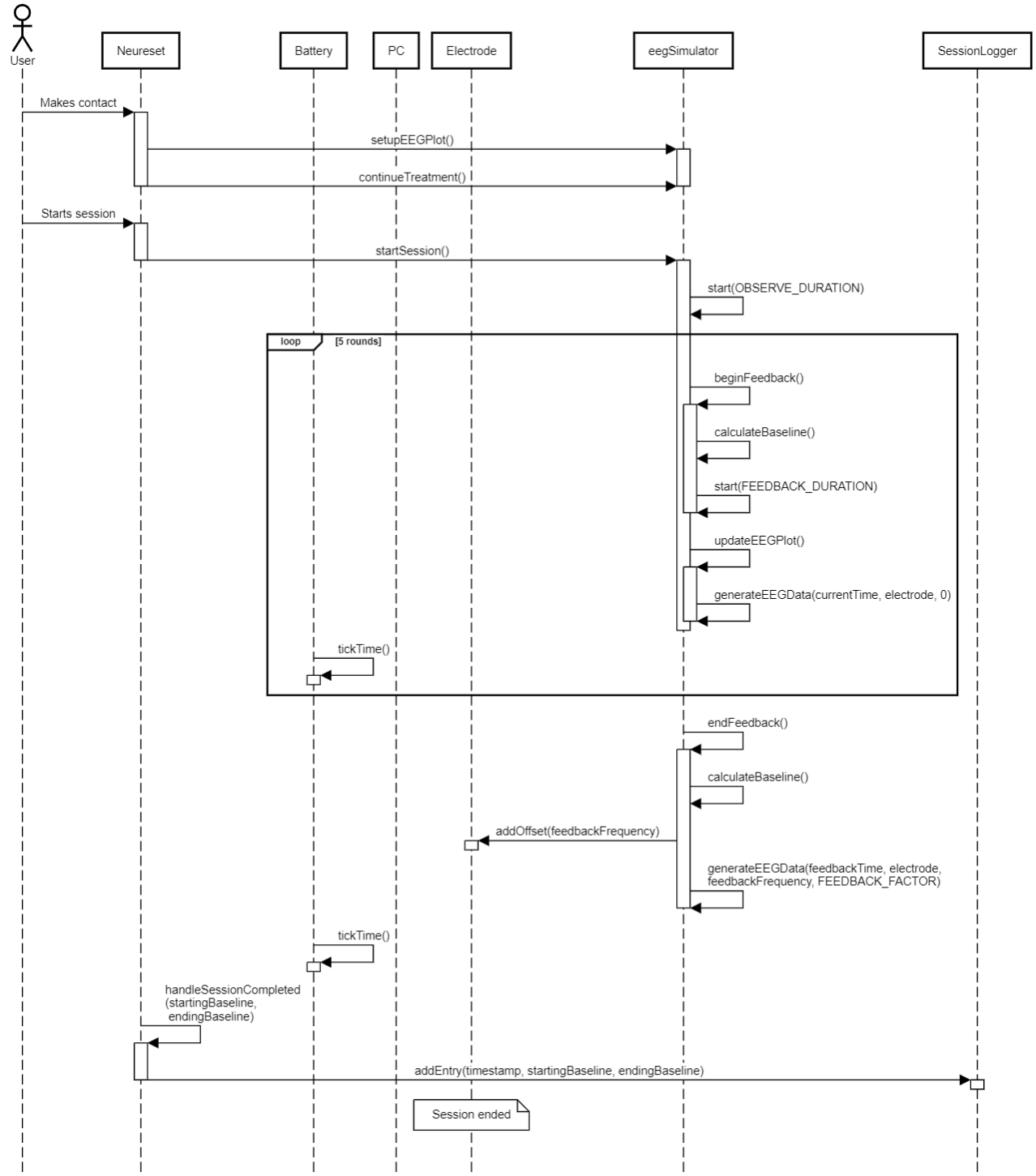

Figure 5 is success scenario of Use Case 4. It is a success scenario when an actor, user, receives treatment. First, the user gets in contact with the device and the corresponding EEG plot and set ups will be in place. Once the user starts the session, the baseline will be calculated, and following offsets will be added every round. After it executes four more times, the final baseline will be outputted, and the final offset will be saved. Then, the session will be added to the log. Note that throughout the treatment, battery is depleted, and it is twice as fast as when not in session.

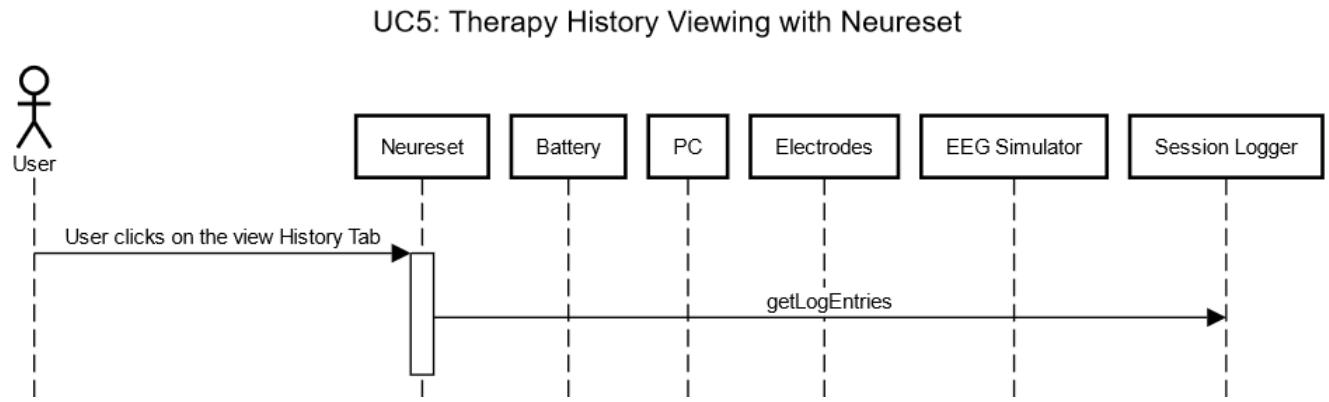


Figure 6 is success scenario of Use Case 5. It is a success scenario when an actor/user presses the view History Tab on the Neureset device. The device will go and ask the session logger which will display all the log entries. Here the log entries only display the date and time of treatments which would be the current time of using the machine or the manually set time had the user set their own time. If there are no log entries, then it will just display a blank page. Within this scenario, we assume that the user has the device powered on and is on the main menu.

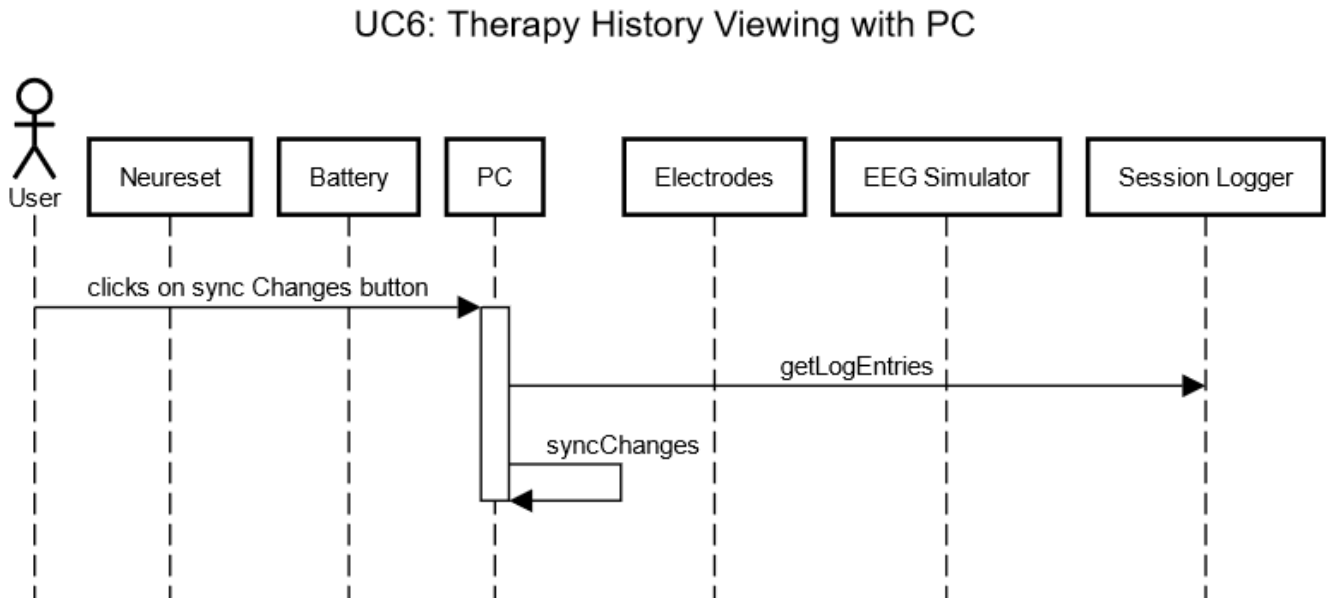


Figure 7 is a success scenario of Use Case 6. It is a success scenario when an actor/user presses the sync change button on the PC. We assume here that the Neureset device is plugged in/mounted onto the PC. The PC will go and ask the session logger for the log entries and then it will sync and display all log entries. Here the log entries have the starting baseline and ending baseline alongside the date and time of treatments which would be the current time of using the machine or the manually set time had the ser set their own time. If there are no log entries, then it will just display a blank page. Within this scenario we assume that the user has the PC powered on.

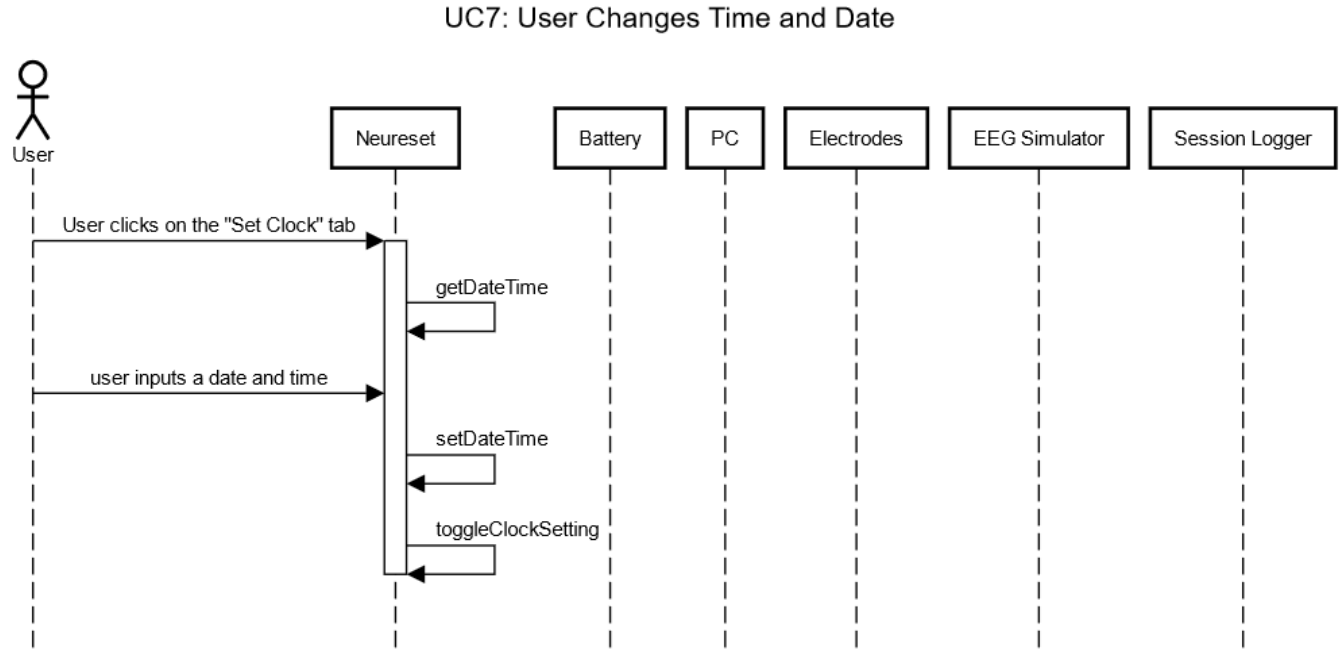


Figure 8 is a success scenario of Use Case 7. It is a success scenario when the actor/user presses on the “Set Clock” tab on the menu on the Neureset device. The Neureset will then get its current date and time.

But then if a user inputs a date and time, the Neureset will set the date and time and then toggle (sync) the clock setting for all other features within the Neureset. Within this scenario we assume that the user has the device powered on and on the main menu.

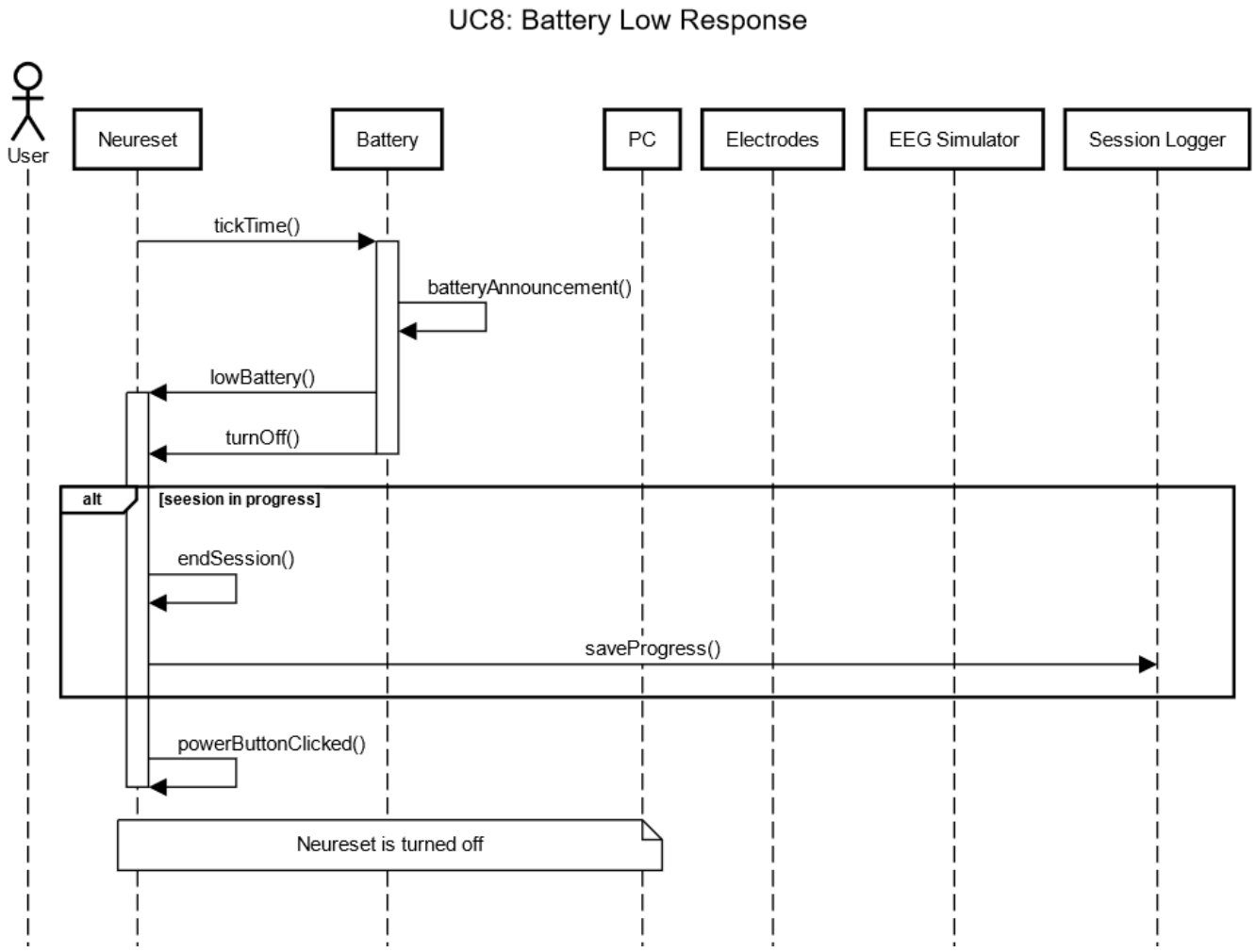


Figure 9 is a success scenario of Use Case 8. It is a safety scenario when the Neureset device is low on battery. Every second, the Neureset will lose battery and more when in use. After a certain threshold of the battery being low, it will announce that the battery is low through audio and text in the device. Eventually, the battery will run out and the device will turn off. If the device is in session while it runs out of battery, it will try to end the session and save progress before powering off.

Note that battery lasts 180 seconds at 100%. When not in session, the battery life goes down by 1 every second and during the session, it is doubled to 2 every second. If the device is off, battery stays constant.

UC9: Connection Lost between Electrodes and the Device

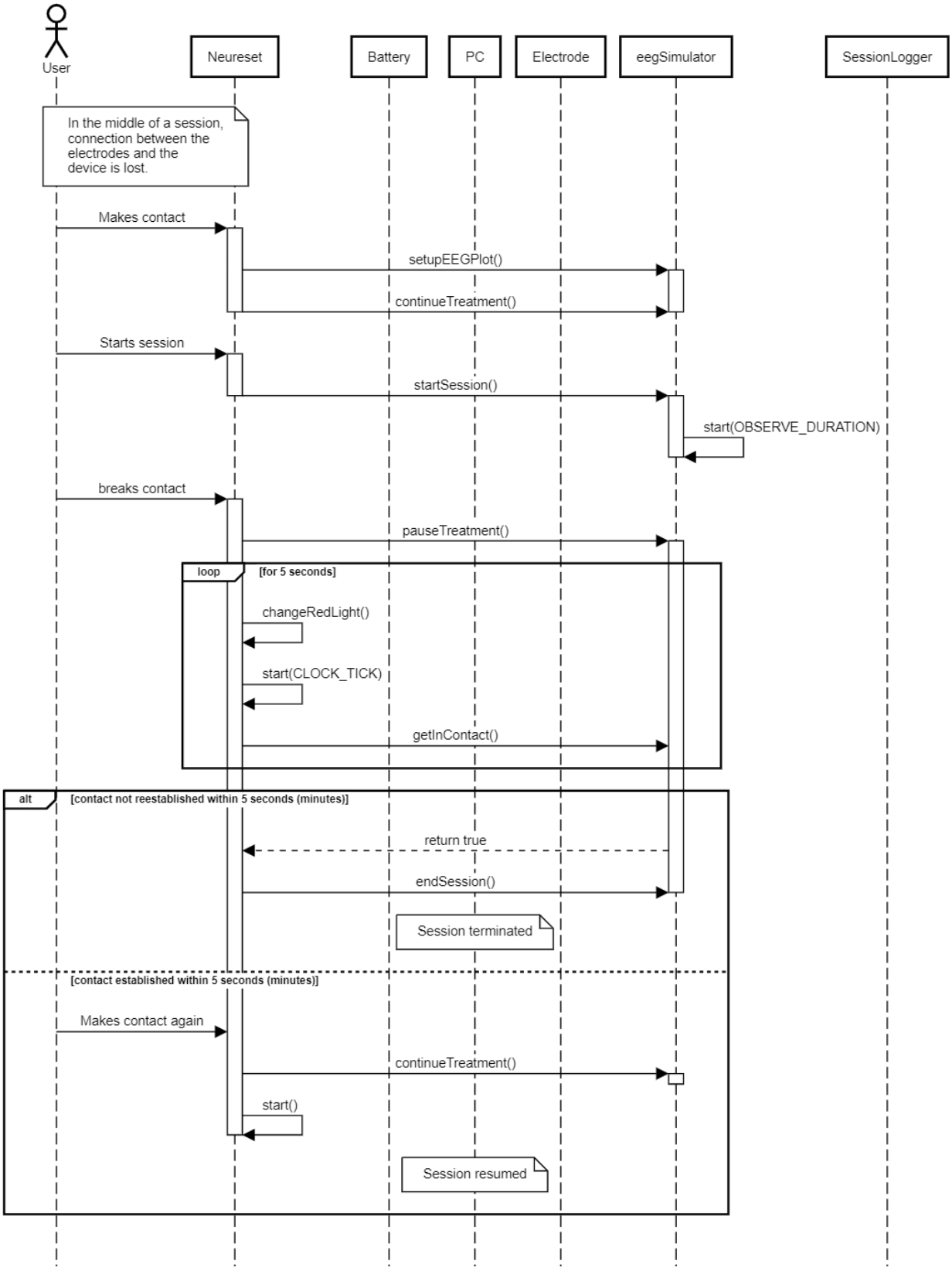

Figure 10 is a success scenario of Use Case 9. It is a safety scenario when the connection between the user and the device is lost. When the user breaks contact with the device in the middle of a session, the treatment is paused, and the red light starts flashing. If the contact is reestablished within 5 seconds (5 minutes in real device), the treatment is resumed. However, if contact is not reestablished within 5 seconds, the session will terminate and it will go back to the main menu, where it prompts the user to contact the device.

## State Diagram

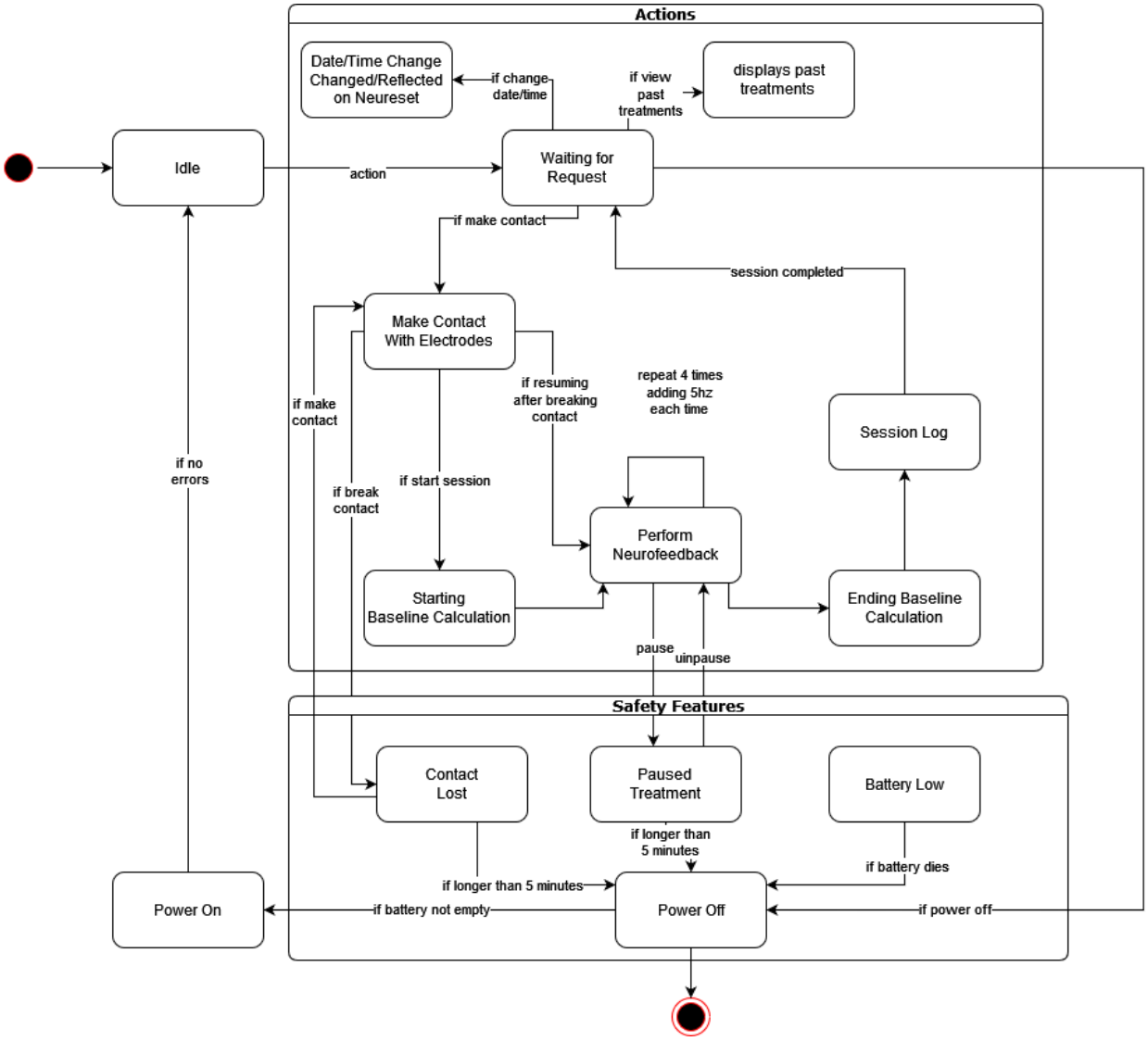


Figure 11 is a state diagram for the Neureset Device for most of the functionalities other than the session log. Note that “Date/Time Change”, and display of “past treatments” can be viewed except when in session.

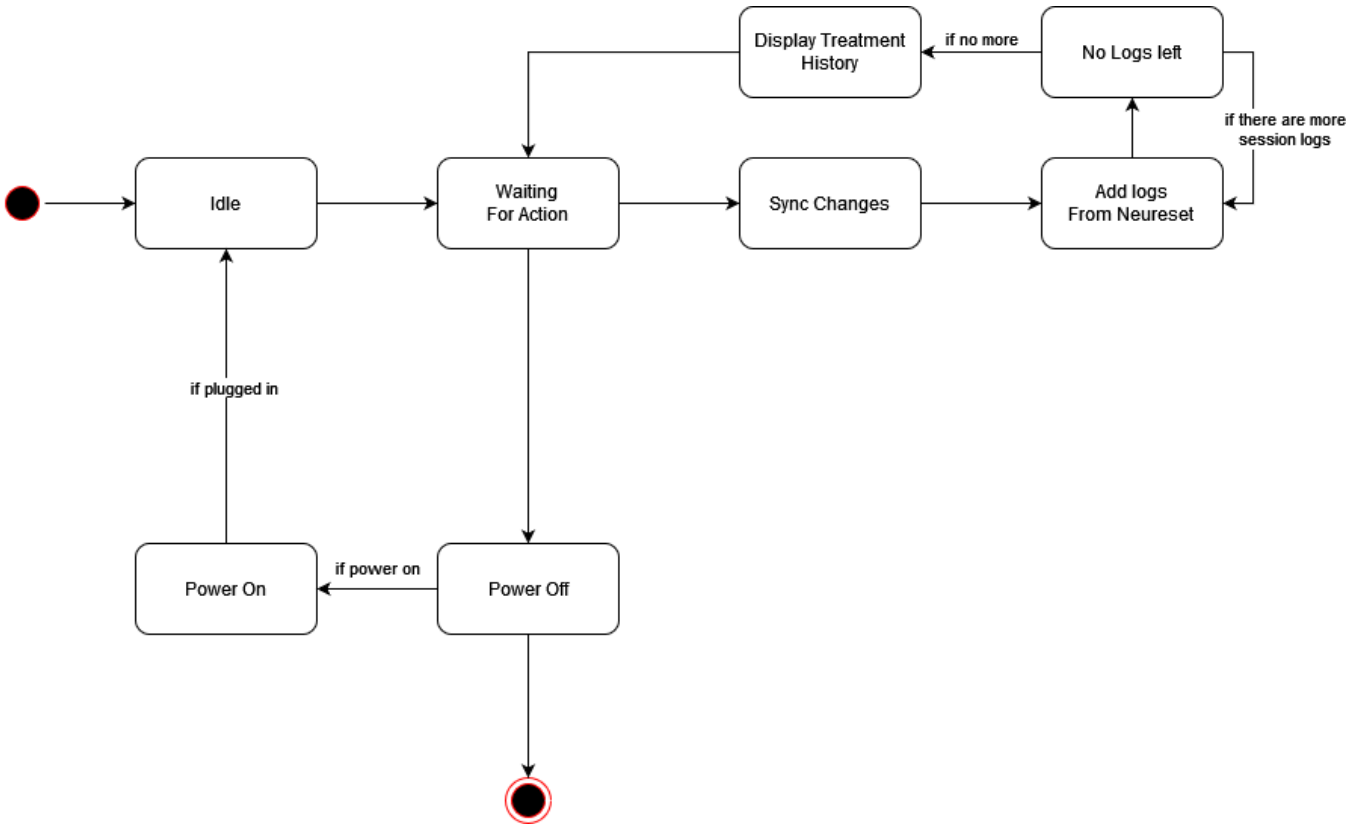


Figure 12 is a state diagram for the connected PC. It will sync session logs and display treatment history on the PC screen.

# Traceability Matrix

| ID | Requirement | Related Use Case | Fulfilled By | Tested by | Description |
|---|---|---|---|---|---|
| 1 | The Neureset interface contains buttons, display, and electrodes. | N/A | Mainwindow.ui | Run the simulator in Qt. View the UI that is titled "Neureset Device" | Reconstructed the physical Neureset device system using QT's built in UI framework. |
| 2 | The PC interface has a button to sync changes and a display for past treatments. | Therapy History Viewing with PC (UC 6) | pcwindow.ui | Run the simulator in Qt. View the UI that is titled "Connected PC" | Reconstructed a PC interface that has a **QPushButton** that simulates synchronizing the **PC Window with the NeuresetController** device.<br><br>This is done because the PCWindow does not have direct access to the **SessionLogger** object, and must rely on interfacing with the **NeuresetController** to extract the **LogEntry data objects**. |
| 3 | The application battery level is dependent on time and whether the device in is session or not. | Turn Device On (UC 1) Battery Low Response (UC 8) | Battery, Neureset Controller | Run the simulator in Qt. In the "Neureset Device" UI, view the battery in the bottom right when the device is turned on – will visually go down as well as gives updates for every 10% decrease in the output logs | Used QTimer to track time passed, whenever time passes, the battery is decreased starting from 3 minutes going down every second. If the device is off, battery level will not change, if on but not in session, will go down by one and if in session then the battery will go down twice as fast. |
| 4 | Treatment cannot start unless user | User Treatment (UC 4) | Mainwindow.ui Electrode, EEG simulator, Neureset Controller | Run the simulator in Qt. In the "Neureset | Tracks if contact is made through the "make contact" / "break contact" button which will be toggled to one or the |

| | | | | | |
|---|---|---|---|---|---|
| | has contacted the electrodes. | | | Device" UI, view how the start session button cannot be pressed till contact is made by the user by pressing the make contact button. | other (based on if contact is made). If contact is not made, button to start a session will be disabled and the user will not be able to start a session. |
| 5 | Treatment displays the progress of the treatment with a bar and a timer. | User Treatment (UC 4) | Mainwindow.ui<br>Neureset Controller<br>EEG Simulator | Run the simulator in Qt. In the "Neureset Device" UI, first make contact and then start a session. View how when a session is started, a progress bar and timer will appear displaying the progress of the treatment as well as the time remaining in the treatment. | The **QTimer::singleShot()** API provided by the Qt library is used to create treatment sessions.<br><br>Since there are always only one treatment session happening at a given time, the simplistic interface provided by the **QTimer::singleShot** function is a good enough abstraction that satisfies all of our requirements. A singleShot timer is started to track the treatment progress (no need for manual multithreading). |
| 6 | The treatment will only progress when the user has contact with the electrodes. | User Treatment (UC 4) | Mainwindow.ui<br>Neureset Controller,<br>EEG Simulator,<br>Electrode | Run the simulator in Qt. In the "Neureset Device" UI, before the session, the graph will 'flatline' indicating that contact is not | The EEGSimulator keeps track of a list of "**state variables**" like **inContact**, **inSession**, **isFeedback**, **isPaused**, **therapyRound**, etc..<br><br>The **inContact** state variable in particular is used to track whether or not the treatment should progress. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | present. Click 'make contact' button and the electrodes will be in contact with the user and the graph will display waves. | We make use of a **polling mechanism** that polls the status of this state variable on every clock **tick** to ensure that treatment pauses as expected on contact lost. |
| 7 | The treatment will stop progressing if the user has elected to pause the treatment. | User Treatment (UC 4) | Mainwindow.ui Neureset Controller, EEG Simulator | Run the simulator in Qt. In the "Neureset Device" UI, when in a session (same process as above) view how when the pause session button is clicked, the timer, progress bar and treatment will all pause. | Once the break contact in Mainwindow.ui is pressed the EEG Simulator will puaseTreatment(). The QTimer for the session and the treatment bar will stop. The clock is still going, Battery is still depleted, and the Electrodes are still being simulated. |
| 8 | The treatment will stop if the user breaks contact with the electrodes | User Treatment (UC 4) Connection Lost between Electrodes and the Device (UC 9) | Mainwindow.ui Neureset Controller, EEG Simulator, Electrode | Run the simulator in Qt. In the "Neureset Device" UI, when in a session (same process as above) view | A turnary operator is used in `**eegValue = inContact ? generateEEGData(currentTime, electrode, 0) : 0;**` to manually override the EEG signal values to "0" to simulate contact loss (a.k.a flat-lining).<br><br>Once the break contact in Mainwindow.ui is pressed the EEG Simulator will pause |

| | | | | | |
|---|---|---|---|---|---|
| | | | | how when the break contact button is clicked, the treatment will stop and wait till contact is restored or a timeout occurs, and the session is stopped. | treatment. The Electrode will stop adding offsets. The Mainwindow.ui will flash red lights for five seconds. If connect is reestablished in 5 seconds, EEG Simulator will **continueTreatment**(), if not it will **endSession**(). |
| 9 | The user can decide to end a treatment if they would like to choose | User Treatment (UC 4) | Mainwindow.ui Neureset Controller, EEG Simulator, Electrode | Run the simulator in Qt. In the "Neureset Device" UI, when in a session (same process as above) view how there is an end session button that is available for users to end a session whenever they want. When clicked, the session treatment will end. | Once the end session in Mainwindow.ui is pressed, the EEG Simulator will endSession(). The Electrode is still being read. |
| 10 | Device supports 7 electrodes and each of the | N/A | Electrode, EEG Simulator, Mainwindow.ui Electrode | Run the simulator in Qt. In the | The 7 electrodes can be viewed from the drop-down menu provided by Mainwindow.ui. The EEG Simulator simulates the |

| | | | | | |
|---|---|---|---|---|---|
| | electrodes have a waveform | | | "Neureset Device" UI, when in a session (same process as above) view how there is a dropdown menu from which electrodes 1 to 7 can be chosen. At all times in a session, the waveform of the chosen electrode is displayed, and users can switch between electrodes to see different electrode waveforms. | seven different electrodes that are configurated by Electrode class. |
| 11 | The waveforms are slightly different and are a combination of frequencies from the five bands: alpha, beta, delta, theta, and gamma | N/A | Electrode,<br>EEG Simulator | Run the simulator in Qt. In the "Neureset Device" UI, when contact is simulated to the device, a waveform will appear. You can change between the electrodes | For each electrode, random Hz values are chosen based on the defined list of channels and their respective ranges (see NeuresetTesting.pdf).<br><br>The specific channels are alpha [8-12Hz], beta [12-30Hz], delta [1-4Hz], theta [4-8Hz], and gamma [25-140Hz].<br><br>This is achieved using QRandomGenerator::global()->bounded() to simulate random brainwave frequencies, while |

| | | | | | |
|---|---|---|---|---|---|
| | | | | using the dropdown and see that they are all different. When a session is started you can also see more specifics on the waveforms printed in the program output. | also reading each electrode's configured public member variables to determine the **MAX_AMPLITUDE** and **MIN_AMPLITUDE** for the sin wave generation. |
| | A realistic EEG waveform is displayed during the treatment process | N/A | Mainwindow.ui Neureset Controller, EEG Simulator, Electrode | Run the simulator in Qt. In the "Neureset Device" UI, when contact is simulated to the device and a session is in progress, there is a waveform displayed as well as a yellow overwritten wave when treatment is in progress showing the waveform during the treatment process. | In the "defs.h" a constant called **NOISE_FACTOR** can be configured to increase or decrease the noise level artificially induced into the system.<br><br>The Electrode class then makes use of QRandomGenerator::global()->generateDouble() * **NOISE_FACTOR** to simulate noise, causing the waveforms to appear more organic and realistic. |

| | | | | | |
|---|---|---|---|---|---|
| 13 | During therapy output to console device activities such as processing input waveform, calculating dominant frequency, delivering the 1 sec feedback at 1/16 of dominant + offset, round 1 of therapy, round 2 of therapy, ..., therapy finished. | N/A | Neureset Controller, EEG Simulator, Electrode | Run the simulator in Qt. In the "Neureset Device" UI, when a session is in progress (following the same steps as above), if you look at the console output during the session, the calculations as well as the step in the treatment are all outputted. | This is done by the EEGSimulator. The EEGSimulator interfaces with the NeuresetController to simulate therapy sessions.<br><br>For example, the EEGSimulator emits a "sessionCompleted" signal to the controller to indicate when a session has completed successfully. |
| 14 | Adjustment of therapy timing for testing: one round in a treatment is 5 sec for analysis instead of 60sec, 1 sec feedback and final analysis of 5 sec for a total of 29sec (4 rounds *6 sec + 5) | User Treatment (UC 4) | Neureset Controller, EEG Simulator, Electrode | Run the simulator in Qt. In the "Neureset Device" UI, when in a session (same steps to start a session), in the console there will be specifics of what is happening in the session which is in sync with the timer and progress bar displayed in the UI. | In "defs.h", there exist several config constants such as **OBSERVE_DURATION** and **FEEDBACK_DURATION** which are respected by the EEGSimulator class.<br><br>The QTimer settings are set dynamically based on these constants.<br><br>To change the timing of each therapy component, simply update "defs.h" and recompile. |
| 15 | Before any neurofeedback is done, the device will have a starting | N/A | Electrode, Neureset Controller | Run the simulator in Qt. In the "Neureset Device" UI, | The EEGSimulator has multiple state variables to keep track of the current state of the simulation. |

| | | | | | |
|---|---|---|---|---|---|
| | baseline calculation for each of the waveforms | | | when in a session (same steps as before), the output in the console will output the baselines for each electrode. | One of the parameters tracked is the baselines of each electrode during the therapy. |
| 16 | After neurofeedback is done, the device will have an ending baseline calculation for each of the waveforms. | N/A | Electrode,<br>Neureset Controller | Like in the test above this, using the same steps to start the session, when the session finishes, the ending baselines will automatically be displayed in the console output. | The EEGSimulator collects the dominant freqs of each Electrode by calling the **getDominantFrequency()** function of each electrode.<br><br>Then, the EEGSimulator executes the **calculateBaseline()** function which uses the simplified formula found in NeuresetTesting.pdf to compute an approximate ending baseline. |
| 17 | After the therapy is complete, the device will log the completed treatment to be recorded. | User Treatment (UC 4)<br>Therapy Viewing with Neureset (UC 5)<br>Therapy History Viewing with PC (UC 6) | Sessionlogger,<br>Neureset Controller,<br>Log Entry,<br>EEG Simulator, | Like in the test above this, using the same steps to start the session, after the session finishes, the ending baselines will automatically be displayed in the console output and recorded into the session logs on the computer which can be viewed in the | A signal-slot mechanism is deployed to handle this one.<br><br>Notice ` **void sessionCompleted(double startingBaseline, double endingBaseline);** ` as the signal definition in EEGSimulator.<br><br>This signal is connected to the NeuresetController's ` **void handleSessionCompleted()** ` slot. |

| | | | | | |
|---|---|---|---|---|---|
| | | | | “Connected PC” UI after syncing it. | |
| 18 | The user can view a history of treatments on Neureset Device | Therapy Viewing with Neureset (UC 5) | Sessionlogger, Neureset Controller, Log Entry, Mainwindow.ui | Run the simulator in Qt. In the “Neureset Device” UI, when treatments are finished, the history of treatments can be viewed by clicking on the “view history” button. This will display history of treatments on the current Neureset Device. | The **NeuresetController** contains an instance of **SessionLogger**, which in turn contains a list: **QVector<LogEntry>**.<br><br>This **SessionLogger** instance serves as the single source-of-truth for both the Neureset device as well as the PC Window view.<br>Note that the Neureset Device only display the timestamp of the completed treatment and none of the starting or ending baselines, |
| 19 | The user can view a history of treatments on connected PC | Therapy History Viewing with PC (UC 6) | Sessionlogger, Neureset Controller, Log Entry, Pcwindow.ui | Run the simulator in Qt. In the “Connected PC” UI after running sessions and syncing them (using the sync changes button), the history of all session ran on the current Neureset Device will be displayed for the user to see. | When trying to access the history of treatments on the PC. PC will first have to sync the logs with the Neureset device. The pcwindow has a **SessionLogger** as well which will turn contains a list: **QVector< LogEntry>**.<br>This **SessionLogger** instance serves as the single source-of-truth for both the Neureset device as well as the PC Window view.<br>Note that in the PC viewing the history of treatments will include the starting baseline and an ending baseline |

| | | | | | |
|---|---|---|---|---|---|
| 20 | The device has a low battery indicator | Battery Low Response (UC 8) | Battery, Mainwindow.uiNeureset Controller | Run the simulator in Qt. In the "Neureset Device" UI, when the battery reaches 20%, there will be an indicator in the bottom left of the UI as well as console output as the battery is decreasing (for every 10%). | The battery **QObject** gets updated by **ticktime()** every second and eventually calls a function **lowBattery()** which has an announcement after a certain threshold which includes an audio and text in the mainwindow.ui. |
| 21 | The device becomes non-functional when the battery level reaches | Battery Low Response (UC 8) | Battery, Mainwindow.uiNeureset Controller | Run the simulator in Qt. In the "Neureset Device" UI, when the battery is at 0% (out of battery), the Neureset Device will automatically turn off and display that there is no battery left in the console as well as the flashing battery symbol indicator. | The battery **QObject** gets updated by **ticktime()** and eventually calls a function **turnOff()** which will turn the device off. The device then goes through its stages of trying to power off the device. If the device is in session while it runs out of battery, it will try to end the session and save the progress before powering off. |
| 22 | The user can manually set the date and time of the Neureset device | User Changes Time and Date (UC7) | Mainwindow.ui Neureset Controller | Run the simulator in Qt. In the "Neureset Device" UI, to change the time from the | The clock starts with a **QDateTime** which is set to the current date and atime that matches the system. The user can manually set date/time with an input of a **QDateTime**. The Neureset will set the new date |

| | | | | | |
|---|---|---|---|---|---|
| | | | | default time, you can go to the “Set Clock” tab and then choose the time you want to set the device to and click “Set Clock”. This will update the time in the clock displayed as well as in the device session history. | and time and then toggle (sync) the clock setting for all other features within the Neureset such as the timestamps for completed treatments. |