Graham Airth, 101046269
Hussien Dabarani, 101042742
Rishiheishan Parameswaran, 101044151
Tom Roberts, 100264321

COMP3004 Final Project
Microcurrent Technology (MCT) Device Simulator
Use Cases and Design Documentation

Table of Contents

Background

Microcurrent technology (MCT) has emerged as a treatment modality that can help to alleviate pain, without the use of medication. A patient applies the electrode(s) of an MCT device to the area of their body where they require treatment -- the device transmits microcurrents through the electrode(s), to the patient's skin tissue, and peripheral nervous system.

Earlier TENS technology uses a stronger electrical current intending to block pain signals (Woollaston, 2014) -- but this can lead to patient habituation and non-responsiveness to the treatment. MCT instead detects the electrical impedance of the patient's skin and modulates its current, thereby preventing habituation. MCT also uses a much weaker electrical current that is better attuned to the body's bio-electrical system.

This report provides use-case, object-oriented analysis, and object-oriented design models of a *simulator* that simulates embedded software used in an MCT device that is not unlike the DENAS-PCM or Avazzia devices.[1] The goal is to simulate the behavior of the embedded software under a range of likely use cases, to better assess possible design issues.

Section 1. Use Case Models

The first use case, UC1, covers the end-to-end behavior of the embedded software at a high-level, during normal patient operation of the device, with a pre-programmed treatment mode.[2] UC2 is an alternative case where the patient instead wants flexible treatment settings that don't relate to a specific health issue. The next two use cases are complementary to UC1 and UC2: power settings (UC3) and display settings (UC4). Use cases for extended non-core functionality (not described here) might include recording (UC5) and using an alarm clock (UC6).

Another use case (UC7) involves the patient or device checking for, or dealing with, non-normal conditions. They include checking the sensor (UC7.1) to see if it is on-skin, or has malfunctioned; checking the battery level (UC7.2); and if the battery is too low, automatically powering-down the device (UC7.3). Lastly, use cases UC8 and UC9 go into more detail on the device's physical components, and user-interface, respectively.

---

[1] Other features of the device (e.g., physical components) help to give context to the simulation.
[2] A use case "captures a contract between the stakeholders of a system about its behaviour" (Cockburn, 2000).

Since the MCT device should provide a wide range of functionality and options for settings, there are many corresponding use cases. Readers are invited to use the handy table of contents to navigate to those use cases and sections that are of most interest to them.

UC1: Normal Operation with Pre-Programmed Settings

UC1	Normal Operation: Pre-Programmed Settings
Description	This is the main top-level use case, and describes normal operation of the device by the patient. It involves only the use of a pre-programmed treatment mode.
Actors	The patient --- has a health condition that results in pain/discomfort.
Triggering Event	The patient: experiences pain, uses the device to treat this pain.
Pre-Conditions	• The device has enough battery power. • The device's physical components work correctly. • Other preconditions outside the scope of this report.[3]
Main Sequence	1. Patient presses the \`on' button. 2. Device boots up. 3. Patient chooses "PROGRAMS" on the device screen, and then a pre-programmed treatment. 4. Patient applies electrode(s) to the affected area. 5. The device's sensors detect feedback from contact with skin 6. The processor assesses the info and sends a control signal to emit the microcurrent waveform that is appropriate for the treatment mode and patient's skin reactivity. 7. The microcurrents are transmitted to the patient through electrode(s) 8. The device display is updated with status of treatment. 9. Repeat steps 6 to 9. In a fully-functioning device, microcurrent would continually adapt to patient state. 10. At the end of use, the patient switches the device off, and it safely powers-down.
Post-Condition	The device is switched off and safely powered-down.
Resulting Event	Patient's pain/discomfort has been reduced (ideally). Battery power has been depleted.
Alternative Scenarios	See UC2.
Comments	This top-level use case includes a range of more specific use cases.
Traceability	See UC1 in Section 4

[3] Other key preconditions could include: (1) a medical practitioner has assessed the patient and has recommended the device; (2) the patient understands how to use the device safely and effectively.

UC2: Treatment With Customized Settings

UC2	Device Use/Treatment With Customized Settings
Description	In contrast to UC1, the patient does not want a pre-programmed treatment for a specific health issue. The patient instead wants to customize certain treatment settings (e.g power, frequency).
Actors	Same as UC1.
Triggering Event	Same as UC1.
Pre-Conditions	Similar to UC1. However, pre-programmed treatment does not meet patient needs.
Main Sequence	-Starts out the same as UC1. -Patient might adjust power setting (UC3). -Patient might customize display (UC4). -Patient chooses from a bunch of alternative treatment menus (see UC9), e.g. Frequency, MED, Screening, Children. -Once a treatment is started, same steps as UC1:6-10, although the patient might continue to adjust settings.
Post-Condition	Similar to UC1. However, might want to save new settings/device state (or not).
Resulting Event	Same as UC1.
Alternative Scenarios	See UC1.
Comments	This top-level use case spans over more specific use cases.
Traceability	See UC2 in Section 4

UC3: Patient Changes Power Settings

A couple of use cases relate to the power level of the device.

UC3.1: Patient changes power level

UC3.1	Normal Operation: Changing Power Level
Description	The patient changes the power level, to adjust to their therapeutic needs or targeted level of battery consumption.
Actors	The patient --- has a health condition that results in pain/discomfort.
Triggering Event	The default power level is too high or low to meet therapeutic needs of the patient, or targeted level of battery consumption.

Pre-Conditions	Similar to UC1, but the default power level is not suitable.
Main Sequence	1. Patient presses the `ON' button. 2. Device boots up. 3. The patient might, or might not, already be using the device for treatment, as in UC1. 4. Patient clicks the RIGHT menu button to increase power, or LEFT menu button to decrease power level. The patient can hold these buttons down, to change level more quickly. 5. Patient continues with normal operation, as in UC1.
Post-Condition	Device is operating at the power level required by the patient.
Resulting Event	The patient can now continue the treatment process and apply electrode(s) to affected area(s). Battery usage
Alternative Scenarios	Certain built-in treatments could automatically change the power level, temporarily (e.g. Minimum Effective Dose, Children settings). Or could use Economy mode (UC3.2).
Comments	None.
Traceability	See UC3 in Section 4

UC3.2: Patient switches to economy mode

UC3.2	Normal Operation: Patient switches to economy mode
Description	The patient changes the device to economy mode to extend battery life.
Actors	The patient --- has a health condition that results in pain/discomfort.
Triggering Event	The patient wants to extend the battery life of the device.
Pre-Conditions	Similar to UC1, although the battery level could be depleted.
Main Sequence	1. Patient presses the 'on' button. 2. Device boots up. 3. Patient selects "Economy" on the device screen. 4. Device switches to economy mode: screen brightness and power level reduced.
Post-Condition	
Resulting Event	Device consumes less battery power. Therapeutic effect might be reduced.
Alternative Scenarios	Patient doesn't use Economy mode. Could instead adjust power level (UC3.1).

Comments	NOT IMPLEMENTED, however, would be a simple extension to the current code implementation, whereby the power level and screen brightness are reduced.
Traceability	N/A.

UC4: Patient Changes Display Settings

In Use Case 4, the patient has powered-on the device, and the device is in working order. However, the patient would like to modify display settings, to improve their user experience. Apart from the display, functionality is not affected.

UC4.1: Patient changes colour scheme

UC4.1	Normal Operation: Patient changes colour scheme
Description	The patient changes the visual colour scheme of the device.
Actors	The patient.
Triggering Event	If the patient feels the colour scheme of the device to be not visually appealing, he or she will attempt to change the colour scheme to their liking.
Pre-Conditions	Device is turned on.
Main Sequence	1. The patient will navigate to the Settings menu using the traversal buttons 2. Once in the settings menu, will traverse to the Colour selection 3. The patient will change the colour scheme accordingly based on the available colours for the device
Post-Condition	The patient will be satisfied with the device's colour scheme in a way that is visually appealing for them.
Resulting Event	The colour scheme of the device has been changed.
Alternative Scenarios	No colour scheme options.
Comments	Try the colour menu on our device.
Traceability	See UC4 in Section 4

UC4.2: Patient changes device language

UC4.2	Normal Operation: Patient changes device language

Description	This use case describes the process of the patient attempting to change the language of the device.
Actors	The patient.
Triggering Event	If the patient is using the device in a language that is unknown to them, he or she will attempt to change the device's language to a language that they know.
Pre-Conditions	Device is turned on, and is in working order.
Main Sequence	4. The patient will navigate to the Settings menu using the traversal buttons 5. Once in the settings menu, will traverse to the Language selection 6. The patient will change the language to their liking accordingly based on the available languages
Post-Condition	The patient will be satisfied with the device language in order for them to use the device effectively.
Resulting Event	The language of the device has been changed and helps the patient to navigate the menu.
Alternative Scenarios	Watch a YouTube video to understand the Russian-language options?
Comments	NOT IMPLEMENTED. But would be a small extra cost to help the device reach a wider market.
Traceability	See UC4 in Section 4

UC5: Patient changes recording settings

Not implemented, and outside the scope of this documentation.

UC6: Patient sets alarm clock

Not implemented, and outside the scope of this documentation.

UC7: Check Device/Exceptions

Use Case 7 deals with checking the functionality of the device, either by the patient or the device monitoring itself. Related to this, it includes exceptions that must be handled.

UC7.1: Check device connection to patient

UC7.1	Check device connection to patient
Description	Patient checks to see if the device is sensing any on-skin contact.
Actors	The patient.
Triggering Event	Patient has tried to apply the device, but isn't sure if it's working.
Pre-Conditions	Device is on, and treatment has started.
Main Sequence	Starts with UC1. Patient has chosen a program/treatment and has applied the electrode, but wants to check that the device is working. Once treatment has started, the patient looks for a message on display to confirm that the device has detected patient skin.
Post-Condition	Message pops-up on display, during on-skin contact.
Resulting Event	The patient can take action if the device hasn't been correctly applied, or if malfunction suspected.
Alternative Scenarios	Device automatically checks whether the electrode has been applied, to monitor treatment.
Comments	None
Traceability	See UC7 in Section 4

UC7.2: Check battery level of the device

UC7.2	Check battery level of the device
Description	Check the battery level of the device.
Actors	The patient.
Triggering Event	Patient wants to check the battery level of the device.
Pre-Conditions	Device is on.
Main Sequence	Patient looks at the screen for an indicator, or looks in Settings.
Post-Condition	Battery level is indicated on the device display.

Resulting Event	Patients can decide whether to continue, reduce power level, or shut-off and charge the battery.
Alternative Scenarios	Device automatically checks the battery to monitor for sufficient battery power.
Comments	This could be implemented in different ways. One could have an on-screen indicator, or a Settings menu option (or both).
Traceability	See UC7 in Section 4

UC7.3: Battery Level is Critically Low: Auto-Shutdown

UC7.3	Auto-Shutdown
Description	Device does an auto-shutdown once it reaches the minimum usable battery level.
Actors	The patient and the device itself.
Triggering Event	The battery level goes below 3%.
Pre-Conditions	The battery level has reached a critically-low level. The device must be able to detect this event (see UC 7.2).
Main Sequence	1. The patient uses the device for his or her own use for treatment. 2. As the device is being used, the battery level gradually decreases in percentage. 3. When the battery level goes below 3%, the device undergoes auto-shutdown. 4. If the patient was in the process of a treatment, the treatment gets cancelled.
Post-Condition	The device is completely turned off.
Resulting Event	If in the middle of treatment, treatment gets cancelled. The patient must charge the battery to some capacity in order to use the device again.
Alternative Scenarios	If the device encounters a malfunction or performance issues and freezes, the device will go into auto-shutdown, and restart, to prevent damage.
Comments	The device can be charged by either plugging it in, or replacing the battery.
Traceability	See UC7 in Section 4

UC8: Physical Components

Use Case 8 relates to the physical components of the device.

UC8.1: Battery

UC8.1	Battery Insertion
Description	This use case describes the input of a battery and the importance of the battery.
Actors	The patient.
Triggering Event	Insert the battery to the device. Device is new, and no battery is inserted, or the battery requires replacement.
Pre-Conditions	Without a battery, the device would not function.
Main Sequence	Insert battery for a functional device for the patient to use.
Post-Condition	Device works, and the patient can use it for their needs.
Resulting Event	Able to use the device and its functions, lights, buttons, etc.
Alternative Scenarios	No battery, and device only plugs in.
Comments	Battery operation provides more convenience.
Traceability	See UC8 in Section 4

UC8.2: On/Off Button

UC8.2	Power On/Off Button
Description	This use case describes turning the device on and off
Actors	The patient
Triggering Event	Pressing the Turn on/off button
Pre-Conditions	The Turn On/Off button is built-in for use
Main Sequence	1. When pressed (Turn On) while off, will turn on the device 2. When pressed (Turn Off) while on, will turn off the device
Post-Condition	Device either turned on or off depending on status of the device

Resulting Event	When turned on, device is ready for use When turned off, device saves battery life
Alternative Scenarios	Button could be used to reset the device when holding button for more than 5 seconds
Comments	Have to press button to enable device functionality.
Traceability	See UC8 in Section 4

UC8.3: Display (Visual Indicator)

UC8.3	Display (Visual Indicator)
Description	This use case describes the display of the device.
Actors	The patient
Triggering Event	Turning the power on by pressing the Turn On button
Pre-Conditions	Pre-built in device
Main Sequence	When device is turned on, the GUI will appear on the display, allowing the patient to traverse through the menus.
Post-Condition	Display is used for GUI to traverse through the functions
Resulting Event	GUI is used and allows the patient to accomplish what he or she desires from the use of the device
Alternative Scenarios	None
Comments	None
Traceability	See UC8 in Section 4

UC8.4: Traversal Buttons

UC8.4	Traveral Buttons
Description	This use case describes the buttons used to traverse through the GUI
Actors	The patient
Triggering	Pressing up, down, left, right, back or select buttons

Event	
Pre-Conditions	Buttons are built-in for use
Main Sequence	1. Up button traverses up on the GUI - Left button traverses left on the GUI - Right button traverses right on the GUI - Down button traverses down on the GUI - Select button traverses to the selected menu of the GUI - Back button traverses to the previous menu of the GUI
Post-Condition	Buttons are pressed, allowing for functionality
Resulting Event	Patient can accomplish what he or she desires from the use of the device
Alternative Scenarios	None
Comments	None
Traceability	See UC8 in Section 4

UC8.5A: Speaker

UC8.5A	Speaker
Description	This use case describes the use of a speaker.
Actors	The patient.
Triggering Event	Functions that require use of the sound system of the device, e.g., alarm, or warning beep.
Pre-Conditions	Speaker pre-built in the device.
Main Sequence	-Could arise out of many other use cases. -E.g. UC1, and treatment is applied for the time limit. -Triggers a warning beep. -Sound plays through the speaker.
Post-Condition	Gives the patient a cue/notification.
Resulting Event	More convenience and safety for the patient.
Alternative Scenarios	Could be customizable.
Comments	Not implemented in our simulator.
Traceability	N/A.

UC8.5B: Patient Changes Volume

UC8.5B	Patient Changes Volume
Description	Patient changes the volume of the device.
Actors	The patient.
Triggering Event	If the patient feels the sound of the device is too quiet or too loud, he or she will attempt to adjust the volume to their liking.
Pre-Conditions	Device is turned on and sound functionality works.
Main Sequence	7. The patient will navigate to the Settings menu using the traversal buttons 8. Once in the settings menu, will traverse to the Volume selection 9. The patient will either increase or decrease the volume accordingly using the left or right buttons
Post-Condition	The patient will be satisfied with the device's volume in a way that is audible for them.
Resulting Event	The volume of the device has been changed
Alternative Scenarios	No sound functionality on the device.
Comments	NOT IMPLEMENTED. Would be important to have, for alarms, warning beeps, etc.
Traceability	N/A.

UC8.6: Connector Jack & Sensors

UC8.6	Connector Jack & Sensors
Description	This use case describes the connector jack and sensor that allows the patient to check their health
Actors	The patient.
Triggering Event	When the device is brought in contact with reactive skin.
Pre-Conditions	Pre-built in device
Main Sequence	1. In the air, the output signal waveform appears 2. The device immediately detects when the electrodes are first placed on

	reactive tissues 3. The device continues treatment reaching optimum electrical characteristics
Post-Condition	The device continues treatment for the reactive tissue
Resulting Event	Reactive tissue is treated
Alternative Scenarios	None
Comments	None.
Traceability	See UC8 in Section 4

UC8.7: Light Sensors

UC8.7	Light Sensors
Description	This use case describes the use of a light sensor
Actors	The patient
Triggering Event	When treatment is being selected and applied
Pre-Conditions	Pre-built in the device
Main Sequence	When treatment is selected and applied, the light sensors turn on and give visual aid to the patient while treatment is undergoing. The light sensors point towards the skin of the patient notifying where the treatment is being applied
Post-Condition	Light turns on
Resulting Event	Treatment is applied with visual assistance, battery levels go down due to light being used
Alternative Scenarios	None
Comments	Not implemented.
Traceability	N/A

UC8.8: USB Interface

UC8.8	Normal Operation: USB Interface

Description	This use case describes the compatibility of the device with a USB.
Actors	The patient.
Triggering Event	Connecting a USB cable supported device to the main device.
Pre-Conditions	No connection between the two devices initially.
Main Sequence	Connect the devices together.
Post-Condition	Device is connected.
Resulting Event	Can analyze, import, or export data between devices.
Alternative Scenarios	Device has no USB connector. However, modern consumers expect to connect with their other devices, and track their data.
Comments	Not implemented in our simulator.
Traceability	N/A

UC9: GUI Traversing

Use Case 9 relates to the traversal of the GUI on the device.

UC9.1: Main Menu

UC9.1	Main Menu
Description	This use case describes the GUI for traversing through the device at the main menu
Actors	The patient
Triggering Event	Turning on the device will prompt the user to the GUI, starting with the main menu
Pre-Conditions	Turn on the device by pressing the Turn On button
Main Sequence	Using the up and down button, can traverse the main menu and make a selection using the select button - Programs - Frequency - Med - Screening - Children - Settings

Post-Condition	Allowed patient to traverse through the main menu and select the desired treatment
Resulting Event	Patient accomplishes what he or she desired from the use of the device
Alternative Scenarios	Can go back to main menu by pressing the Back button
Comments	None
Traceability	See UC9 in Section 4

UC9.2: Programs Menu

UC9.2	Programs Menu
Description	This use case describes the GUI for traversing through the device at the Programs menu
Actors	The patient
Triggering Event	From the main menu, selecting Programs will take the patient to the Programs menu
Pre-Conditions	Turn on the device by pressing the Turn On button and traverse to Programs menu
Main Sequence	Using the up and down button, can traverse the Programs menu and make a selection using the select button - Allergy - Pain - Int. Pain - Bloating - Dystonia - Gyn. Pain - Hypotonia - Head - Throat - Muscles - Constipation - Trauma
Post-Condition	When selection is made, device begins “Applying Program Treatment”
Resulting Event	Patient accomplishes what he or she desired from the use of the device
Alternative Scenarios	Can go back to Programs menu by pressing the Back button
Comments	None

Traceability	See UC9 in Section 4

UC9.3: Frequency Menu

UC9.3	Frequency Menu
Description	This use case describes the GUI for traversing through the device at the Frequency menu
Actors	The patient
Triggering Event	From the main menu, selecting Frequency will take the patient to the Frequency menu
Pre-Conditions	Turn on the device by pressing the Turn On button and traverse to Frequency menu
Main Sequence	Using the up and down button, can traverse the Frequency menu and make a selection using the select button - 1.0-99Hz - 10Hz - 20Hz - 60Hz - 77Hz - 125Hz
Post-Condition	When selection is made, device begins “Applying Frequency Treatment”
Resulting Event	Patient accomplishes what he or she desired from the use of the device
Alternative Scenarios	Can go back to Frequency menu by pressing the Back button
Comments	None
Traceability	See UC9 in Section 4

UC9.4: MED Menu/Treatment Option

UC9.4	Minimum Effective Dose (MED) Menu/Treatment Option
Description	This use case describes the GUI for traversing through the menu to the Minimum Effective Dose (MED) menu
Actors	The patient
Triggering	From the main menu, selecting MED will take the patient to the MED menu

Event	
Pre-Conditions	Turn on the device by pressing the Turn On button and traverse to MED menu
Main Sequence	Start with UC1, but apply MED Treatment instead of Program-menu treatment.
Post-Condition	When selection is made, the device begins applying the minimum effective dose treatment.
Resulting Event	Patient accomplishes what he or she desired from the use of the device
Alternative Scenarios	None
Comments	None
Traceability	See UC9 in Section 4

UC9.5: Screening Menu/Treatment Option

UC9.5	Screening Menu/Treatment Option
Description	This use case describes the GUI for traversing through the device at the Screening menu
Actors	The patient
Triggering Event	From the main menu, selecting Screening will take the patient to the Screening menu
Pre-Conditions	Turn on the device by pressing the Turn On button and traverse to Screening menu
Main Sequence	Applying Screening Treatment
Post-Condition	When selection is made, device begins “Applying Screening Treatment”
Resulting Event	Patient accomplishes what he or she desired from the use of the device
Alternative Scenarios	None
Comments	None
Traceability	See UC9 in Section 4

UC9.6: Children Menu

UC9.6	Children Menu
Description	This use case describes the GUI for traversing through the device at the Children menu.
Actors	The patient.
Triggering Event	From the main menu, selecting Children will take the patient to the Children menu.
Pre-Conditions	Turn on the device by pressing the Turn On button and traverse to Children menu.
Main Sequence	Using the up and down button, can traverse the Children menu and make a selection using the select button: - Up to 1 Year - 1-3 Years - 4-7 Years - 7-12 Years
Post-Condition	When selection is made, the device begins “Applying Child Treatment”.
Resulting Event	Patient accomplishes what he or she desires from the use of the device.
Alternative Scenarios	Can go back to the Children menu by pressing the Back button.
Comments	None
Traceability	See UC9 in Section 4

UC9.7: Settings

UC9.7	Normal Operation: Settings
Description	This use case describes the GUI for traversing through the device at the Settings menu.
Actors	The patient.
Triggering Event	From the main menu, selecting Settings will take the patient to the Settings menu.
Pre-Conditions	Turn on the device by pressing the Turn On button and traverse to Settings menu.
Main Sequence	Using the up and down button, can traverse the Settings menu and make a selection using the select button:

<table>
<tr><td></td><td>

Brightness
Economy
Recording
Clock
Alarm Clock
Language

English
French
German
Cantonese

Colour

Black
Blue
Red
Green

Volume

</td></tr>
<tr><td>Post-Condition</td><td>When selection is made, the device can change the settings accordingly.</td></tr>
<tr><td>Resulting Event</td><td>The patient can adjust display and other settings, to have a better user experience.</td></tr>
<tr><td>Alternative Scenarios</td><td>Can go back to the Settings menu by pressing the Back button.</td></tr>
<tr><td>Comments</td><td>None</td></tr>
<tr><td>Traceability</td><td>See UC9 in Section 4</td></tr>
</table>

Figure 1: UC1 Use Case Diagram

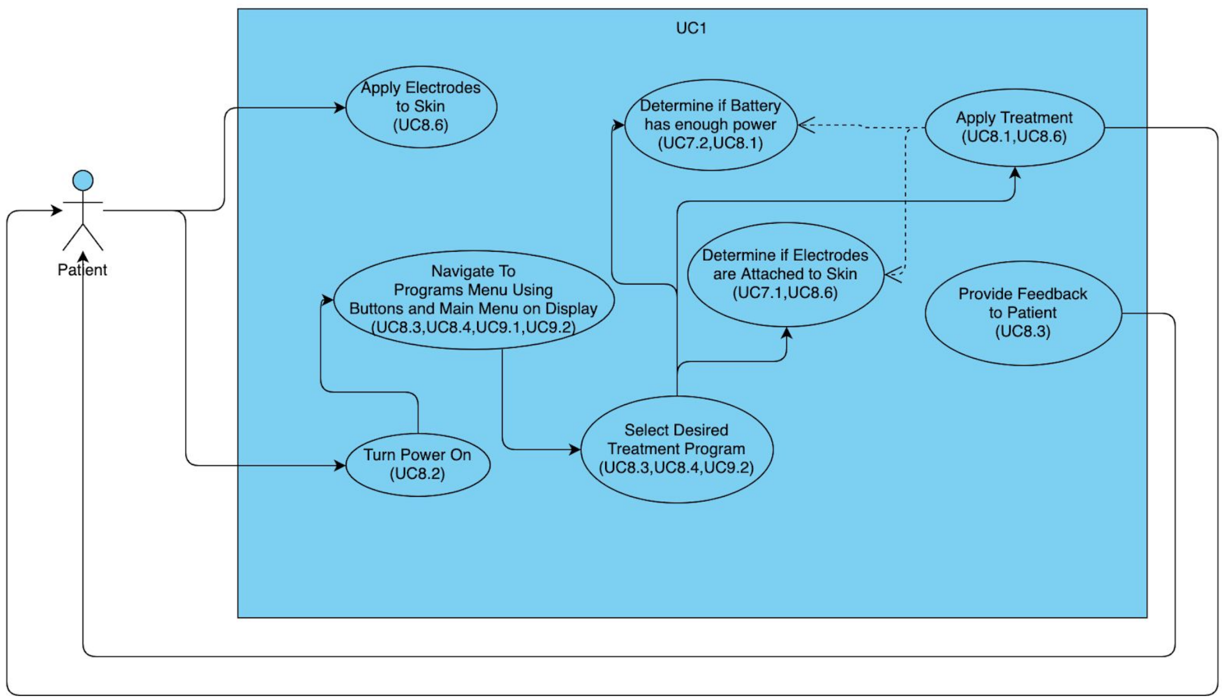


Figure 2: UC2 Use Case Diagram

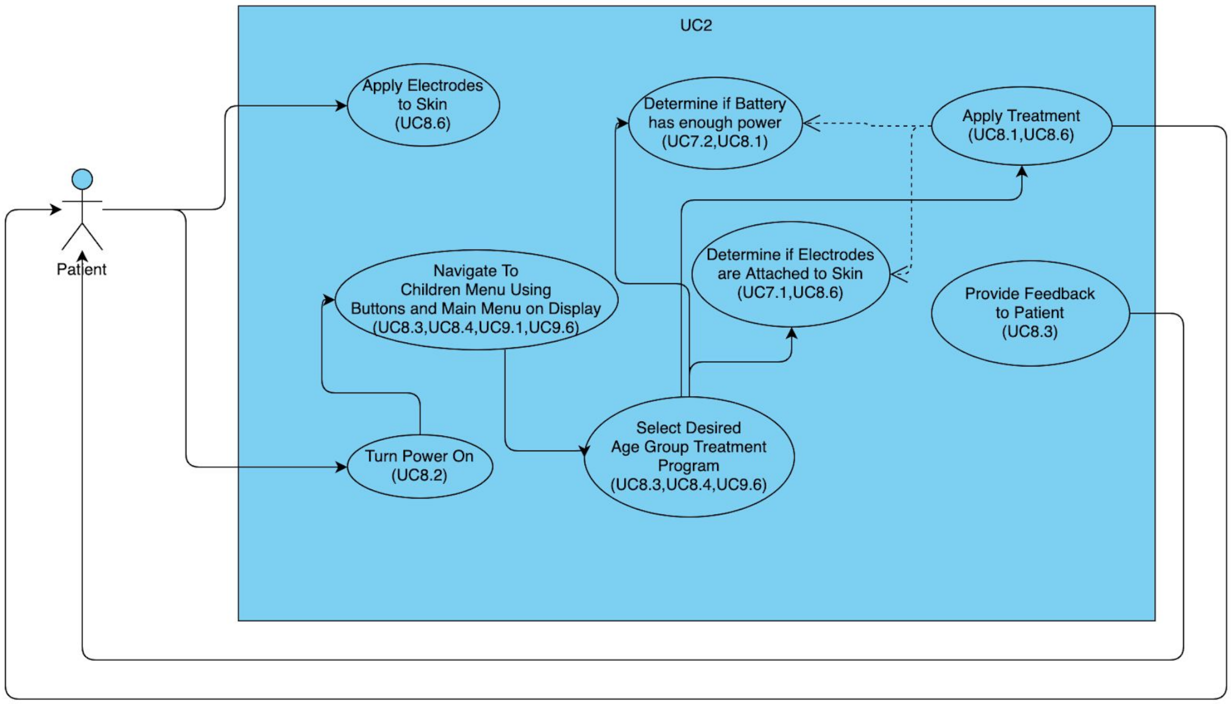


Figure 3: UC3 Use Case Diagram

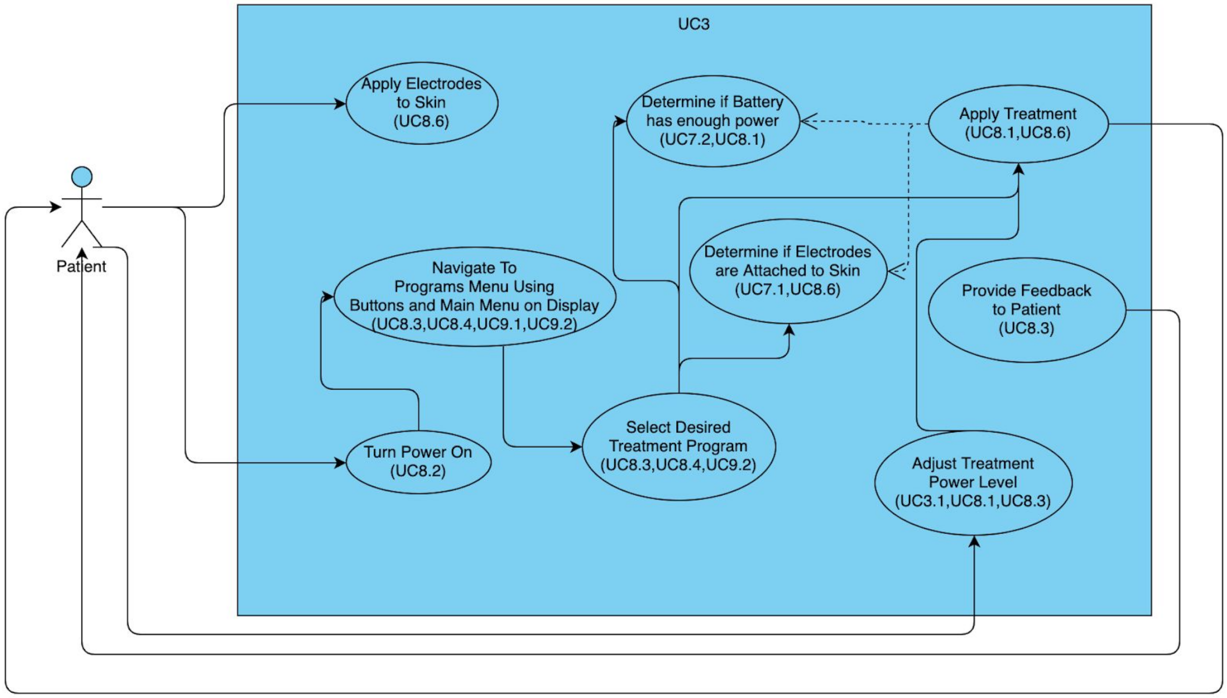


Figure 4.1: UC4.1 Use Case Diagram

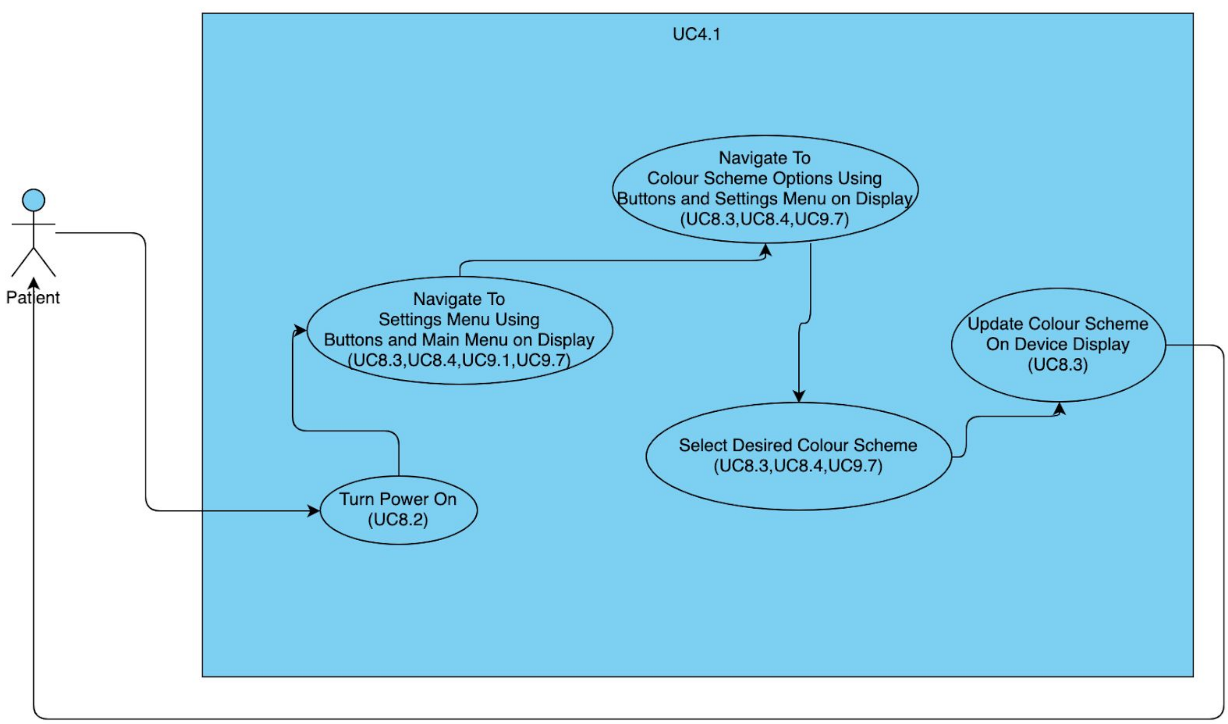


Figure 4.2: UC4.2 Use Case Diagram

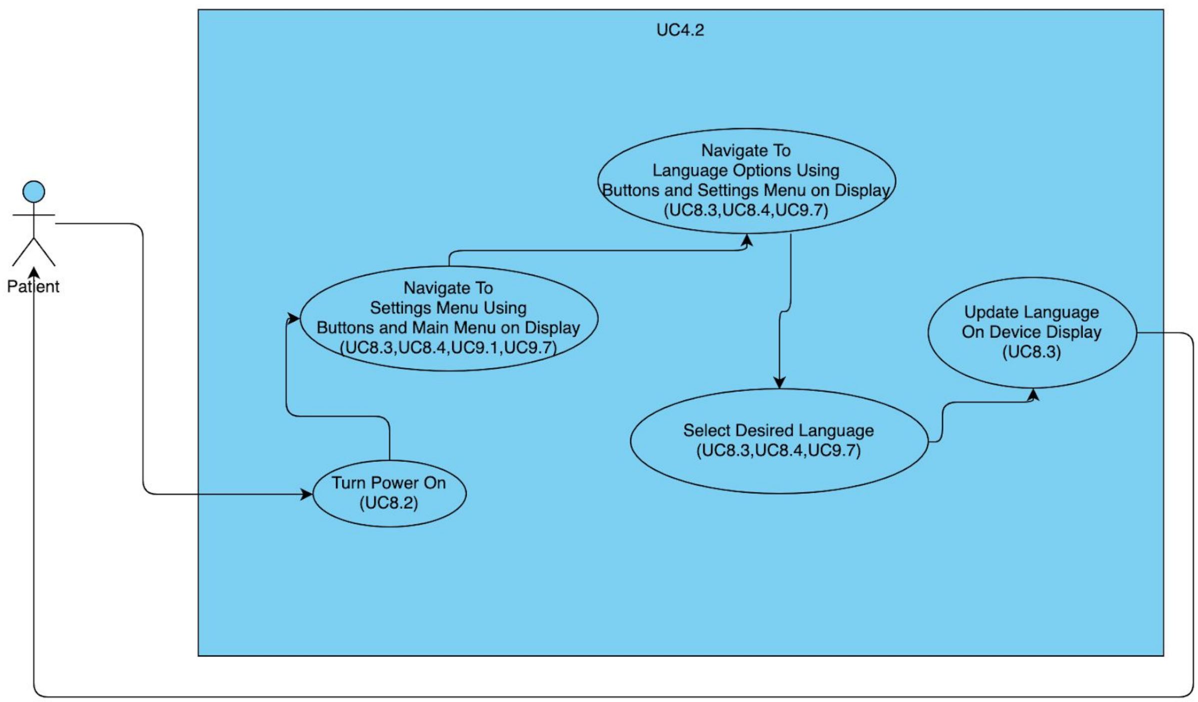


Figure 5: UC5 Use Case Diagram

Not implemented, and outside the scope of this documentation.

Figure 6: UC6 Use Case Diagram

Not implemented, and outside the scope of this documentation.

Figure 7: UC7 Use Case Diagram

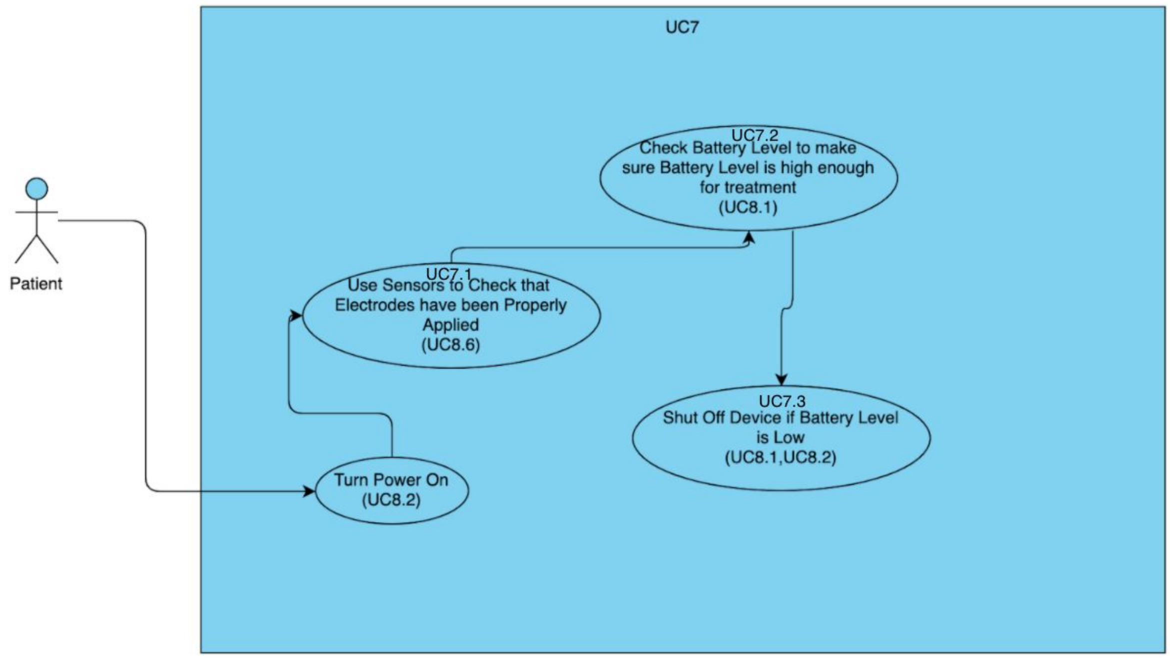


Figure 8: UC8 Use Case Diagram

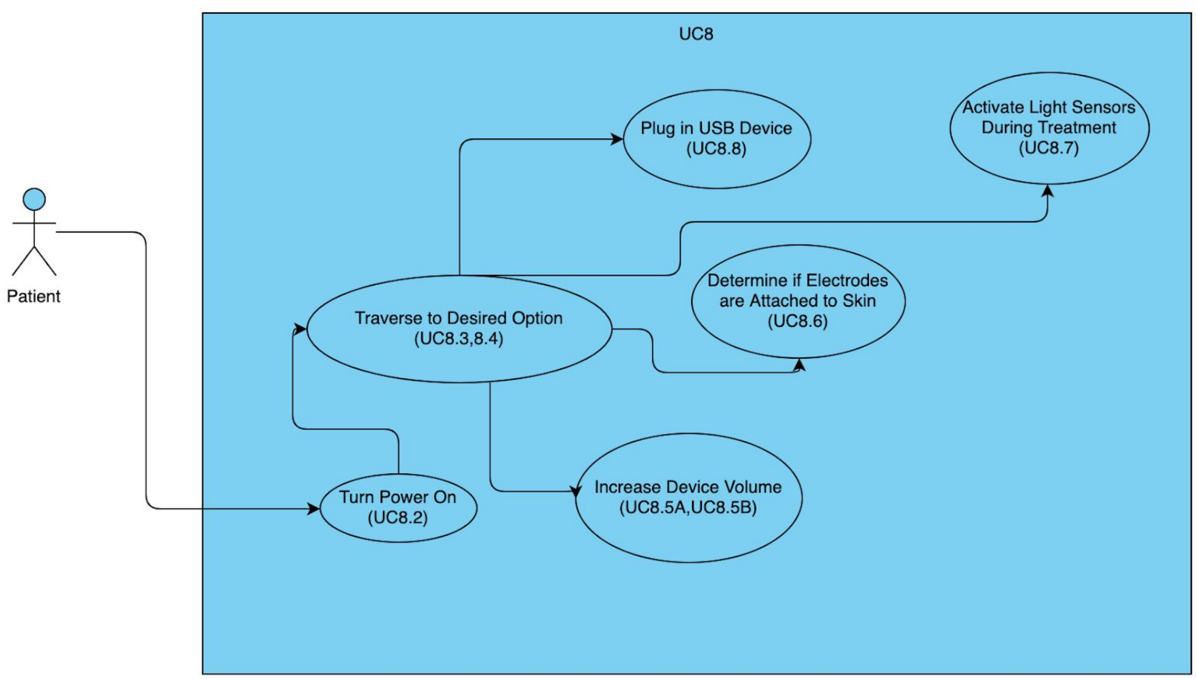
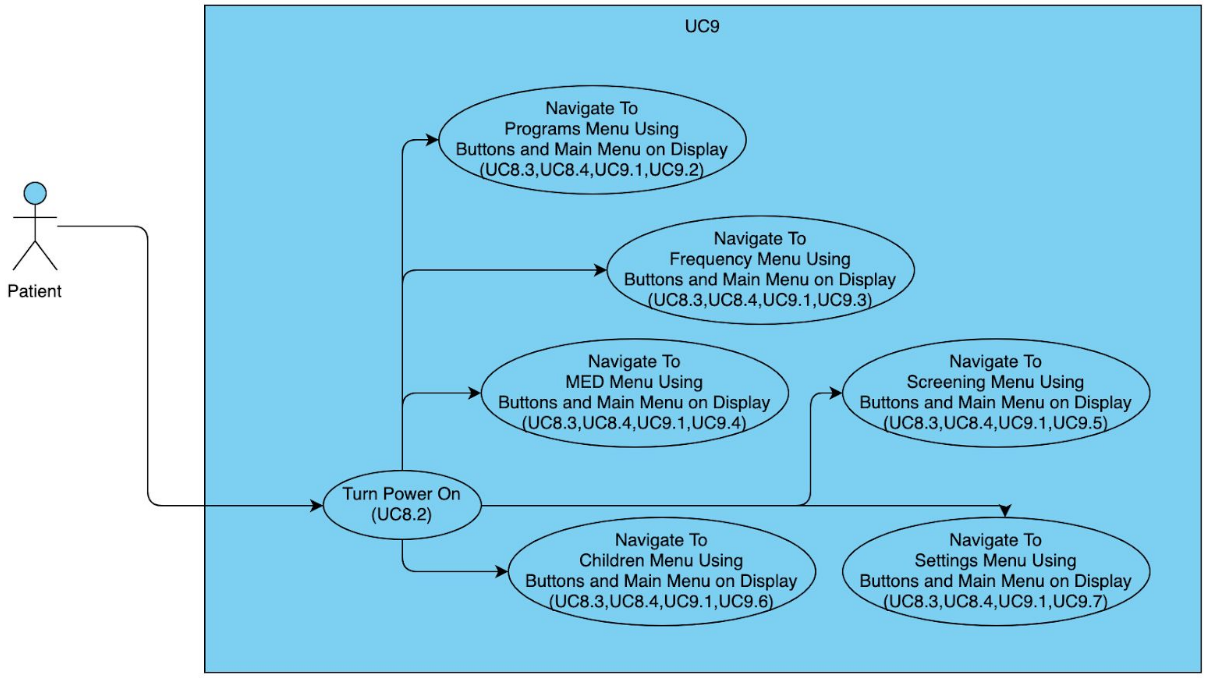


Figure 9: UC9 Use Case Diagram



Section 2. Object-Oriented Analysis (OOA)

The MCT device simulator could be modeled as having a range of classes, illustrated in Figure 10, including a Simulator Control, the Device, Battery, and Electrode. Although these classes are interdependent, each has distinct functionality for which it could help to model separately. The battery and electrode are also distinct physical components of the device, as a whole.

The user/patient is external to the simulator but represents the primary actor that initiates changes in the device's state. Certain user actions, like recharging the battery, or applying the device's electrode to the skin, don't have corresponding 'buttons' on the device itself, and so could be handled by a Simulator Control. The Simulator Control can also close the simulator application -- an action that is distinct from turning the device 'on' and 'off'. The 'on'/'off' actions are key aspects of functionality that we might instead want to simulate within a particular application session.

A Device class models the actual device and user interface that the user/patient would interact with in real-life operation. The Device class could act as a mediator, as a way of coordinating and decoupling the other classes and centralizing device control.[4] The Device should have attributes (some of which are objects, themselves) for the device user-interface, menu options, treatment, and display settings. It also has methods to initialize the device's on-state, get information from its sensor about whether the electrode is 'on' or 'off' the patient's skin, signal the electrode to emit micro-currents and get the battery's power level.

A display class, which is part of the device user-interface, could have brightness and color attributes. There would need to be methods to update the menu display and to set these display attributes.

An Electrode class takes in a set of inputs that describe the treatment mode and then outputs a microcurrent waveform. These inputs include a boolean for whether the electrode is on the patient's skin, as well as the power level and frequency of the treatment. The electrode (or at least, its software class representation) would use some mathematical function(s) to transform the inputs into the waveform output.

The menu consists of a set of sub-menus, current (sub)menu and current-row attributes. The contents of these menus are described in detail in the use cases in the previous section. Methods link the device user-interface (i.e. buttons) with the menu navigation. A menu selection potentially then sets the attributes for the treatment or other device settings.

[4] Distinct from the simulator control discussed earlier.

The Battery class is straightforward, but vital for the device. Its main attribute is its battery (power) level. It has a method to provide its battery level to the Device class and to also handle the event of its battery level being depleted, as the device is operated. The level of depletion should be conditional on the power level of the device's microcurrent output.

Together, these classes should allow us to model/simulate most of the core use cases: pre-programmed and other treatments (UC1, UC2); customized settings (UC3, UC4); checking for, or handling exceptions (UC7); and details on traversing menus to choose specific options (UC9). Certain aspects of the physical device components are also simulated (UC8), e.g. the battery and display.

Additional classes for extended non-core functionality (not described here) could include clock (with the date and time attributes) and more detailed display settings. Other extensions, not pursued in this project, could allow the sensor or electrode to fail, either with some random probability or given input from the Simulator Control.

Figure 10: OOA Class Diagram

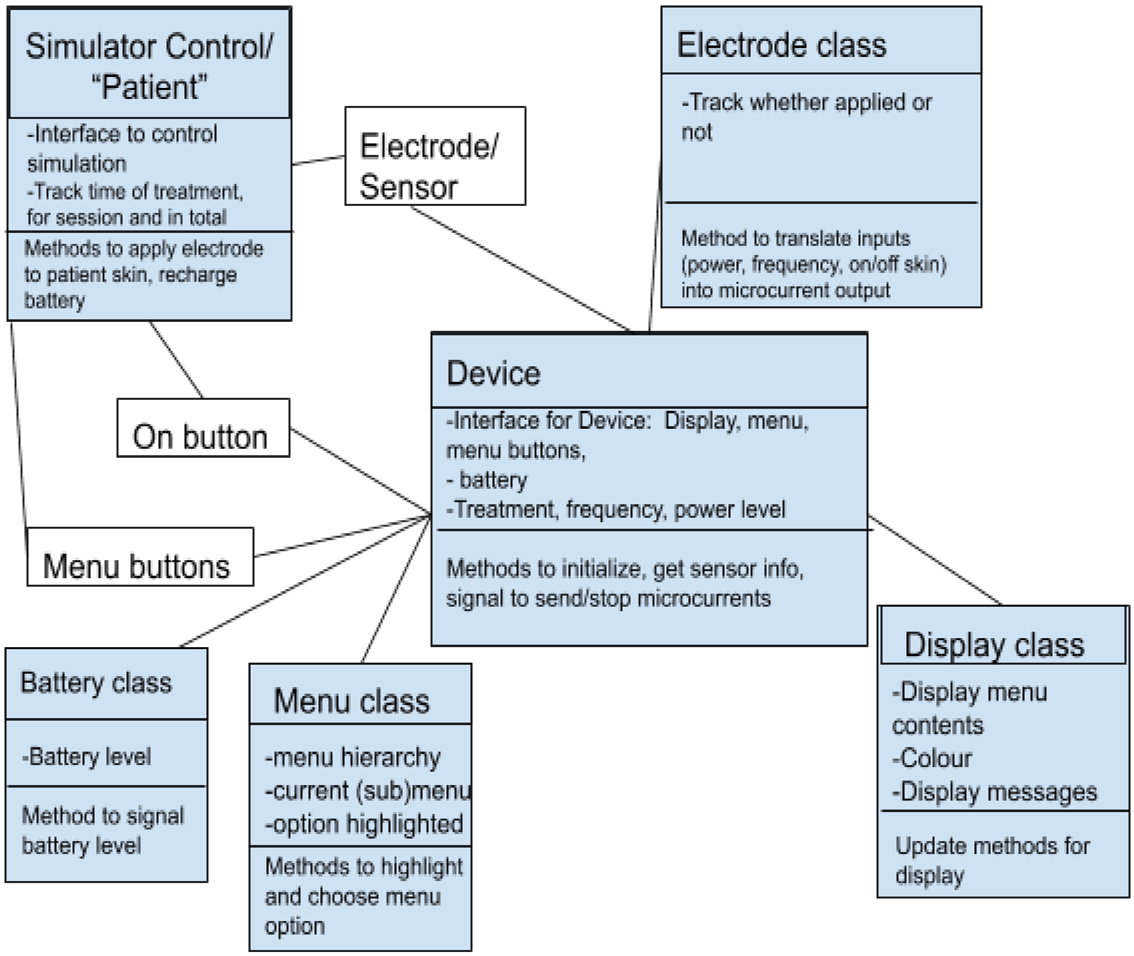


Figure 11: Sequence Diagram for Use Case 1

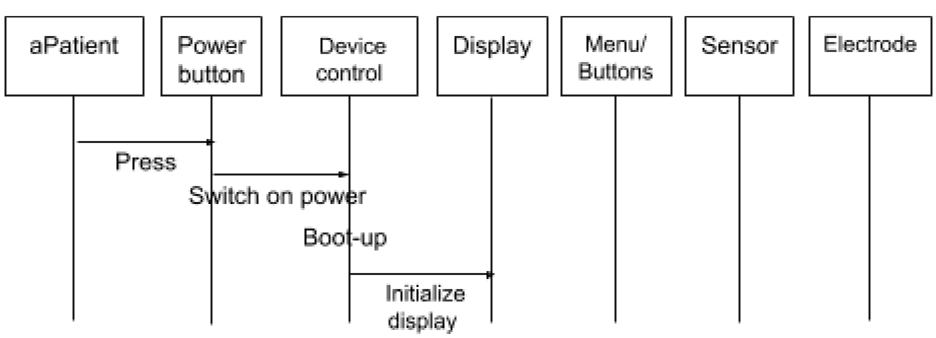


Figure 12: Sequence Diagram for Use Case 1

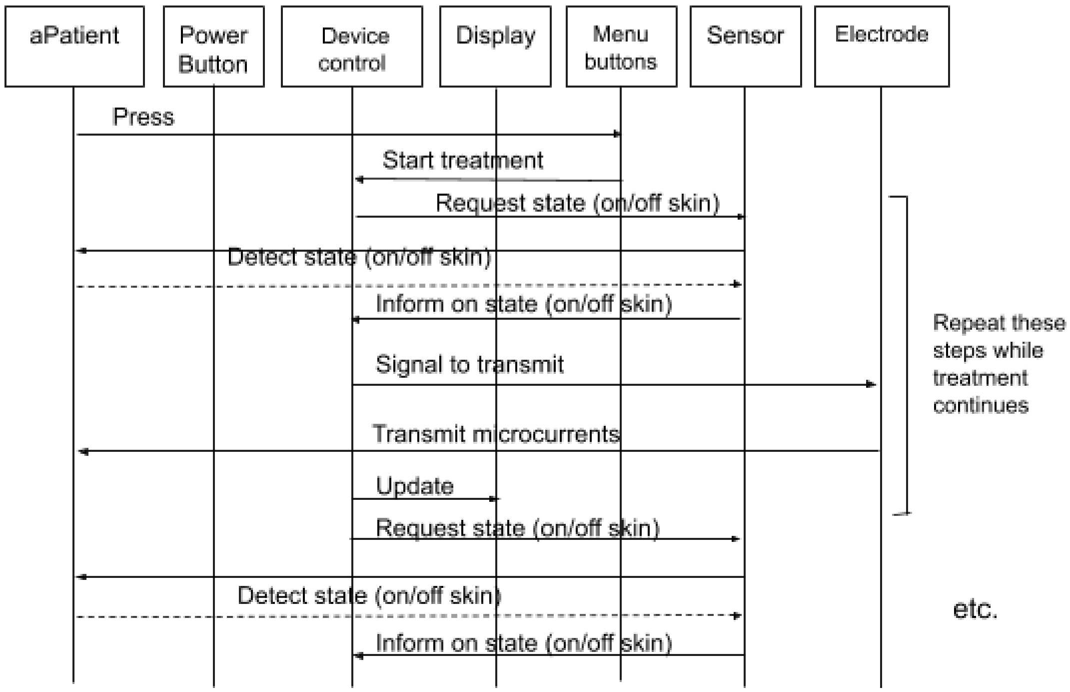


Figure 12: Sequence Diagram for Use Case 7.3

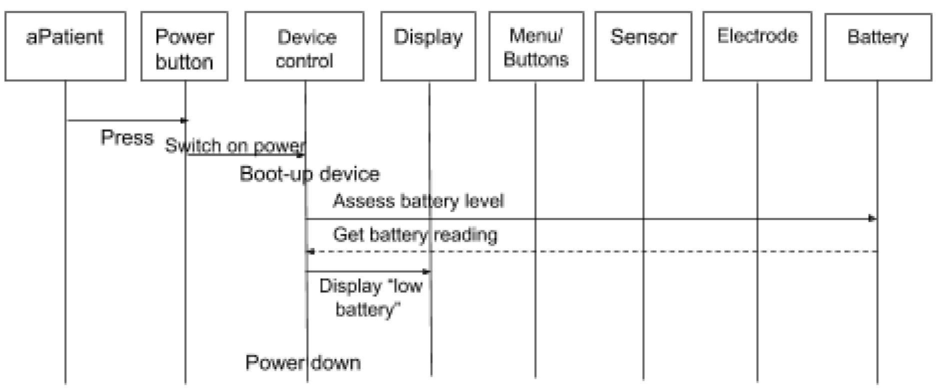
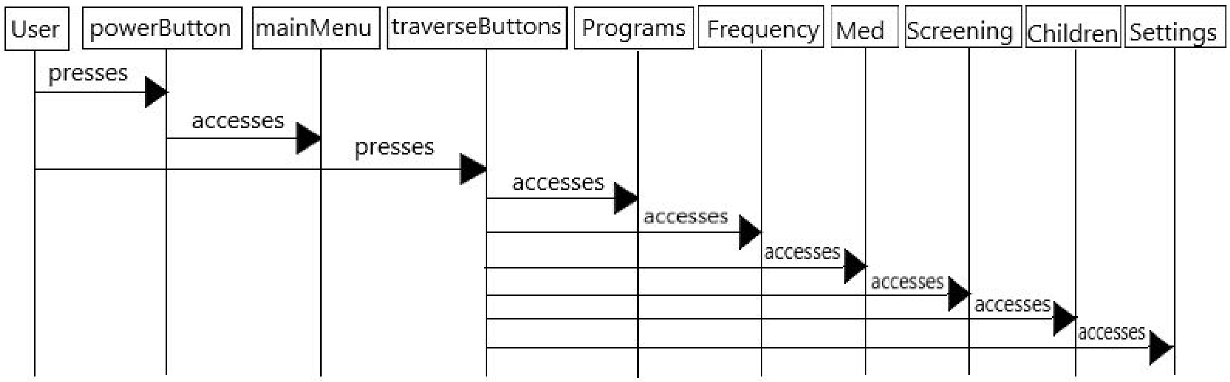


Figure 13: Sequence Diagram for Use Case 9



Section 3. Object-Oriented Design (OOD)

The object-oriented design (OOD) stage extends the OOA to go into more detail on the simulator design, though it is also more focused. Rather than following a waterfall development model, where one step follows another in a linear sequence, we have taken a more agile/iterative approach, so the design has been revised to reflect updates to requirements, as well as practical issues with the implementation.

The project requirements have evolved throughout the implementation phase, which has required adjustments to the OOD along the way --- for example, the requirement to add a timer for on-skin treatment. The design also partly reflects the Qt implementation, since we go into more detail on variable types and methods, in many cases, and because the design has been influenced by Qt's features, such as the Qt event loop, and the Signals & Slots support.

Some features of the OOD:

- Class design diagram, shown in Figure 14
    - Names and types to attributes.

    - Names and return values to methods.
    - More detail on associations.
- The sequence diagram includes method names and parameters (Figure 15)
- An activity diagram can also show how multiple processes fork out from the same initial steps of turning the device on.

Many attribute types shouldn't require much explanation (e.g. *float* for the battery level). A few device attributes define the microcurrent waveform that is outputted by the electrode: *isPowerOn* (bool), *powerLevel* (int), *freqLevel1* (int), *freqLevel2* (int), *isFreqAlt* (bool), and i*sApplied* (bool). Together, these attributes can define a wide range of pre-programmed and custom treatment modes.

Associations between objects are further specified, in the class design. In most cases, there is one instance of a class (e.g. device, device UI, battery, electrode) at a time, resulting in a 1-to-1 association. Some classes could have a specific number of instances (e.g. sub-menus, represented by instances of QStringList) that are greater than one. For this simulator, we only have one user that acts as the patient, and who presses buttons on either the simulator control or the device user interface.

In many ways, the Device class acts as a mediator. It either sends most signals or intermediates between the other classes. It also owns/instantiates the QTimer, which drives the Qt event loop, and the resulting updates to the battery level, electrode output, and LCD time. This can be seen in Figure 14 with the *sendTick()* signal sent by the Device, and the *updateBatteryLevel()*, *updateElectrodeOutput()*, and *updateLCD()* methods that act as slots for this signal.

The intermediation can be seen where the user presses the "Apply Electrode" button, which sends a signal, *signalDeviceApplied(),* that the electrode is on the skin. The device (i.e., through its sensor) receives this signal with its *updateSensorInfo()* slot, and then sends the updated info in the *sendTick(...)* signal. The electrode then updates its output with the *updateElectrodeOutput(...)* slot.

A couple of exceptions to the Device as a mediator are that the Simulator Control can directly signal the battery to recharge, with *signalBatteryCharge()*, and it can also close the simulator application. Further discussion of design patterns is left to the demo.

The sequence diagram in Figure 15 shows method calls for the main part of UC1. There is some simplification, for example, with the navigation through the menu, which would require multiple button presses. One can see calls to *selectOption()*, as choices are made on the menu. Note that in our code implementation, the sensor is no longer a distinct class, and becomes part of the Device class.

Figure 14: OOD Class Diagram

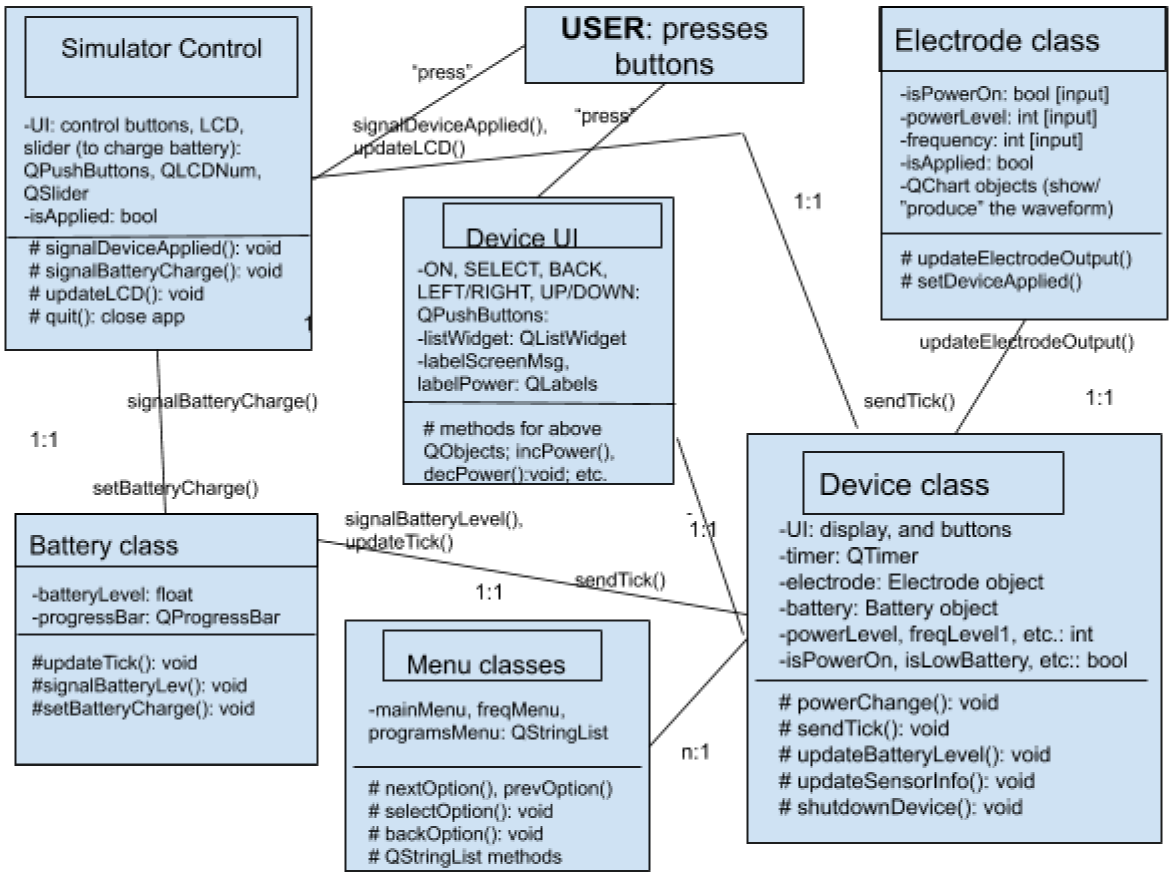
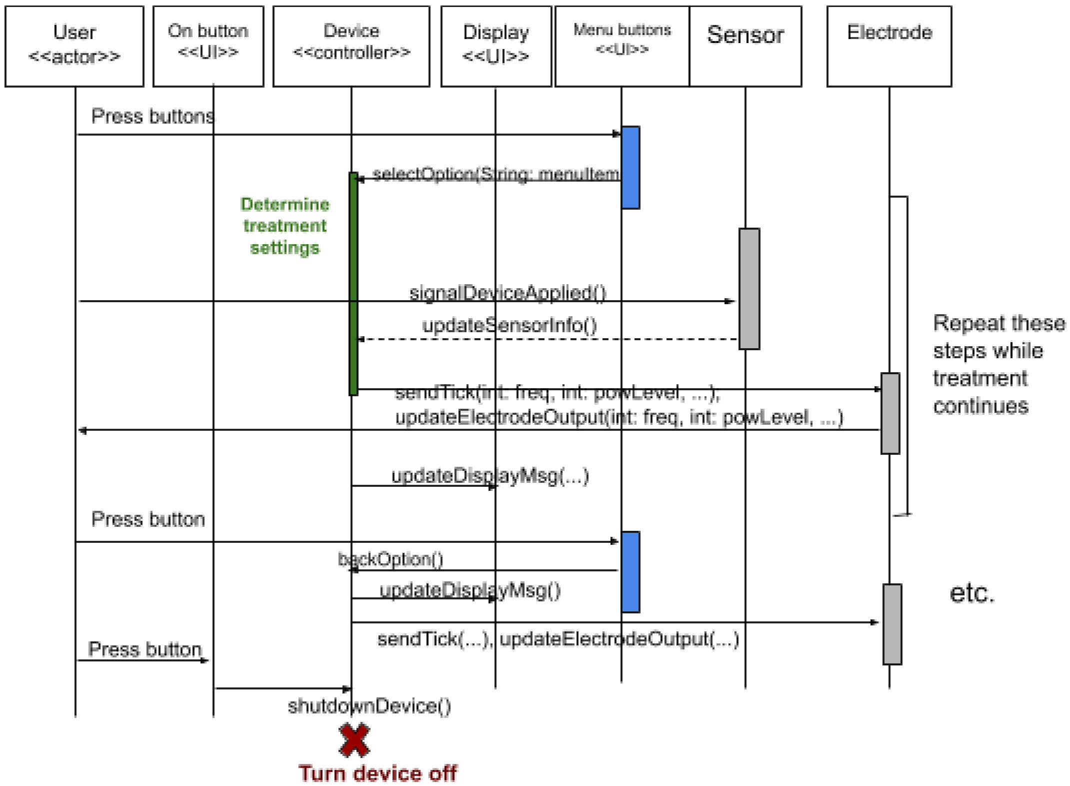


Figure 15: Sequence Design Diagram



The activity diagram in Figure 16 helps to illustrate how the various use cases are related and branch off from one another at specific junctures. The use cases UC1 and UC7.3 are identical up until the stage of assessing battery power. UC2, involving a change in the treatment mode, branches out and reconnects to UC1. UC3 to UC6 (not shown) would appear as similar side-paths.

UC7.3, where battery power is insufficient, forks off after the battery reading finds low power, and the device powers down, after displaying a notification. Conversely, the other use cases involve displaying the menu or (if applicable) continuing with the current treatment. UC7.1 enters into UC1 when checking the sensor for info -- if on-skin contact is detected, the electrode(s)'s output is modulated accordingly.

Figure 16: Activity Design Diagram

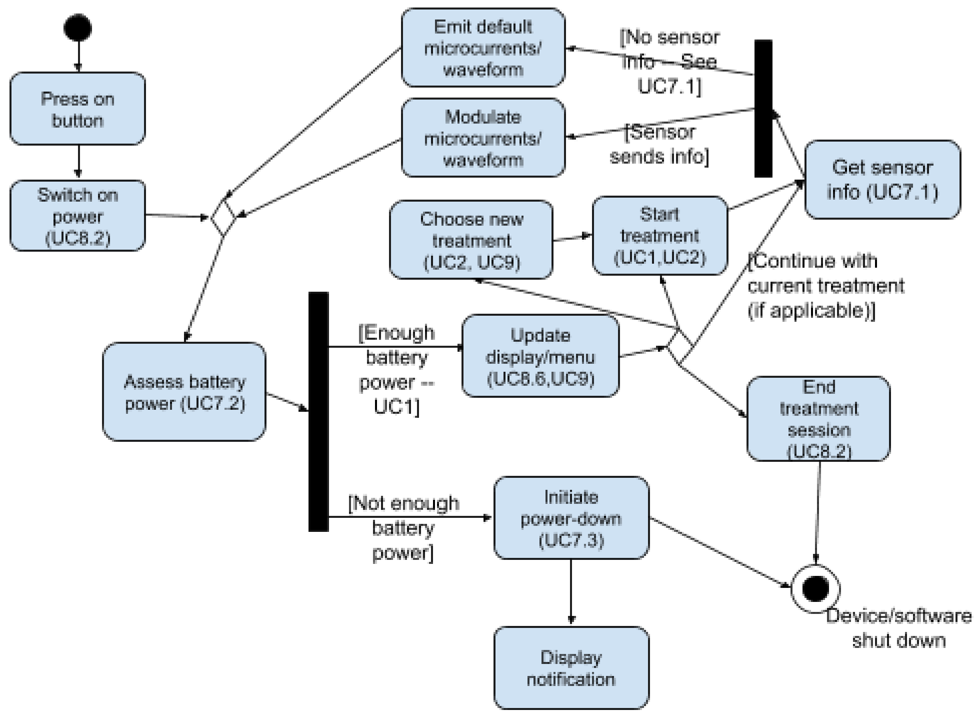


Figure 17 shows the simulator layout, where one can see the simulator's main classes with their own respective windows, while Figure 18 explains the correspondence (visually) of the simulator's GUI to the use cases described in section 1.

Figure 17. Simulator Layout

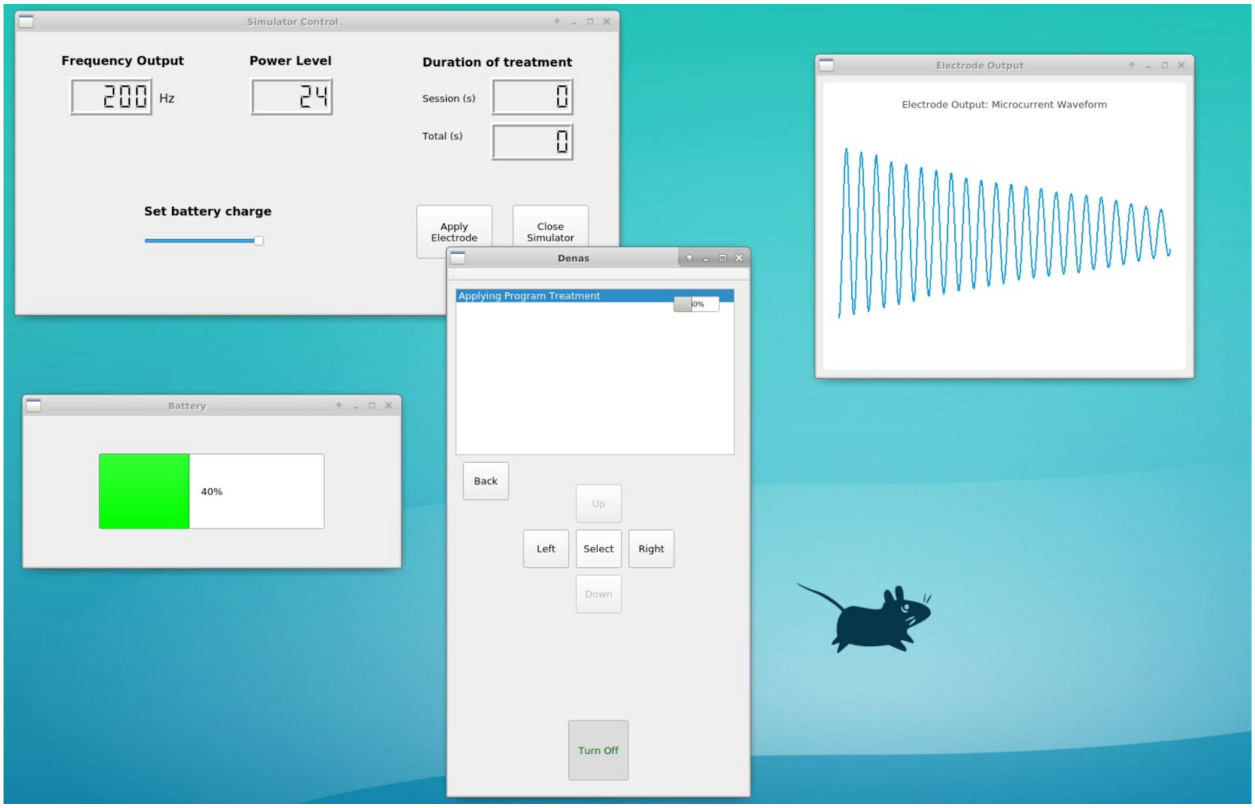
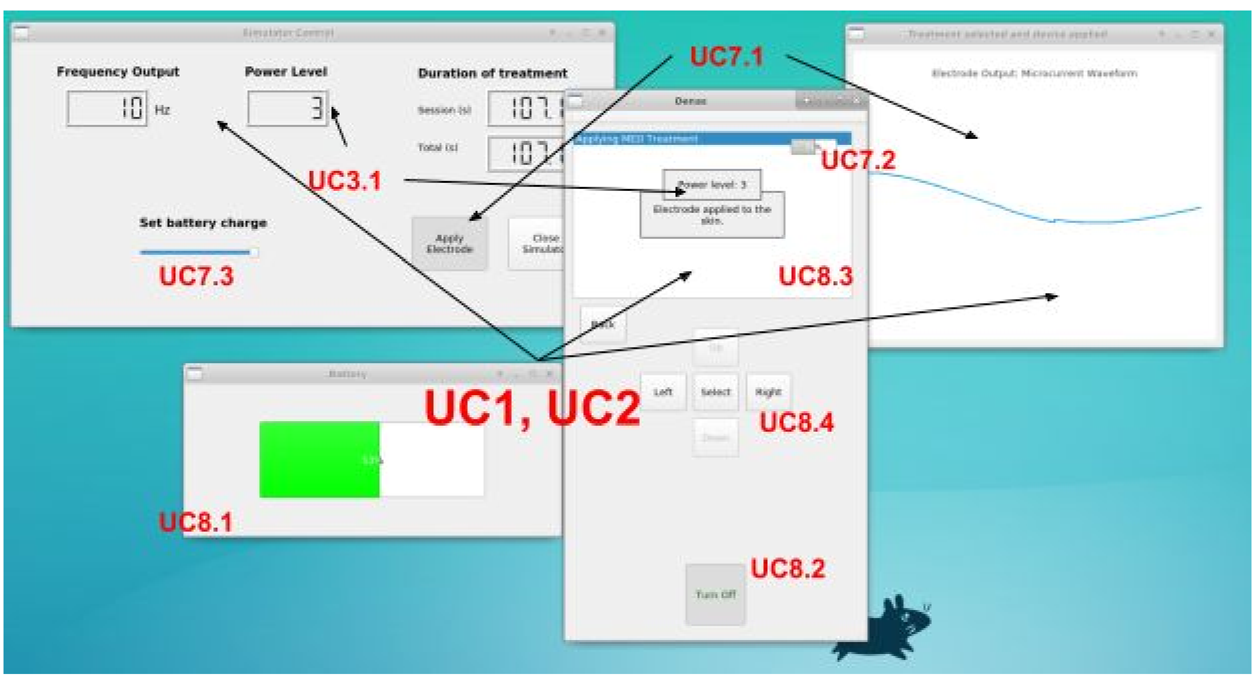


Figure 18. Simulator and Use Case Correspondence



Section 4. Traceability Matrix

Requirement Number	Requirement Description	Test Case ID	Status
UC1	Normal patient operation of the device, with pre-programmed treatment	TC1	Test Passed
UC2	Patient changes the treatment mode, and administers treatment	TC2	Test Passed
UC3	Patient customizes power settings	TC3	Test Passed
UC4	Patient customizes display settings	TC4	Test Passed
UC7	Device monitors the electrode sensors and battery level	TC5	Test Passed
UC8	Device components are used during device operation	TC6	Test Passed
UC9	Navigate through all menus	TC7	Test Passed

Test Case ID	Test Case	Test Steps	Test Data	Expected Result
TC1	Verify Normal Operation	1. Power on device 2. Apply device pads to patient's affected area 3. Select treatment program 4. Device verifies pads are connected properly and battery level is sufficient 5. Treatment Applied	Successful	Successful administration of treatment
TC2	Verify Treatment	1. Power on device	Successful	Successful

	Change	2. Apply device pads to patient's affected area 3. Select alternative treatment program 4. Device verifies pads are connected properly and battery level is sufficient 5. Treatment Applied		administration of alternative treatment
TC3	Verify Treatment Customization	1. Power on device 2. Apply device pads to patient's affected area 3. Select treatment program 4. Device verifies pads are connected properly and battery level is sufficient 5. Treatment Applied 6. Patient increases power level of treatment	Successful	Successful customization of treatment
TC4	Verify Settings Customization	1. Power on device 2. Select settings menu 3. Select languages menu 4. Select new language for device	Not fully implemented (menu exists but alternate languages not entered)	Change the language of the device.
TC5	Verify Monitoring of Sensors and Battery	1. Power on device 2. Attempt and fail to apply normal treatment 3. Apply device pads to patient's affected area 4. Attempt to apply normal treatment	Successful	Treatment applied only if the battery and sensors are charged and attached.
TC6	Verify Operation	1. Test that all of the	Successful	All buttons

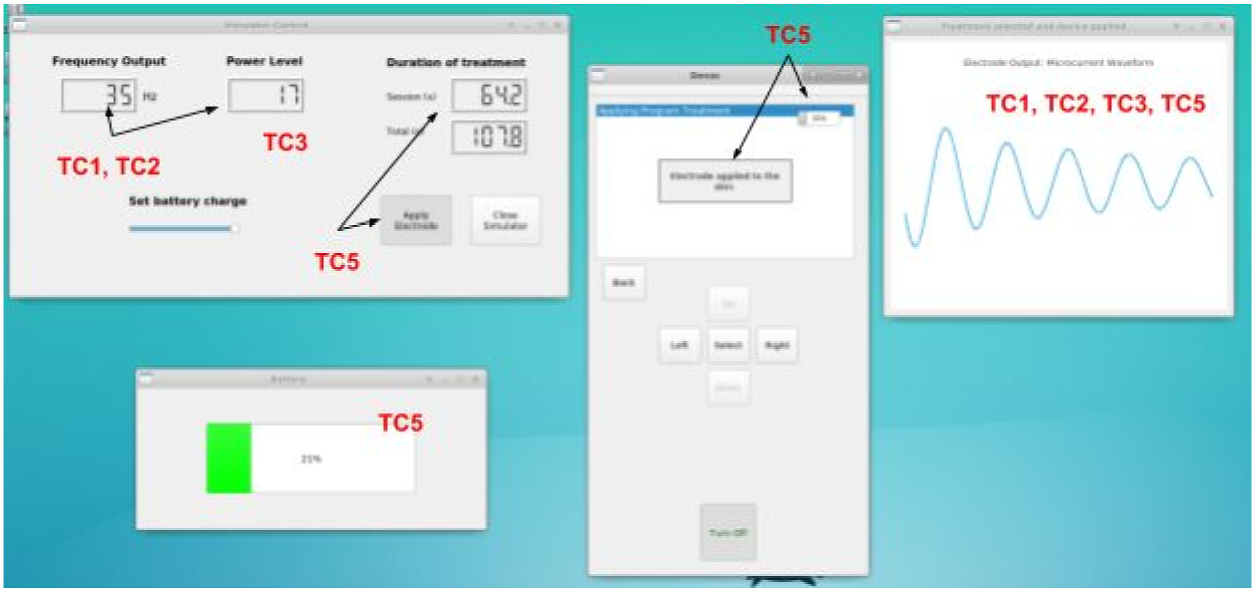
	of Components	physical buttons are functioning on the device.		are working as designed.
TC7	Verify Menu Navigation	1. Enter each menu and submenu of the device	Successful	All menus are accessible for the user

To complement the testing, for our simulator we have combined the simulated device with a set of real-time monitoring tools that show device inputs and outputs, in the form of a Simulator Control window, as well as QWidgets for the battery and electrode output. The correspondence of the test cases to these tools is shown in Figure 19.

This enables the user to easily confirm the following, while simulating the device's usage:

- Battery level
    - Device monitors the battery level.
    - Device responds to critically-low battery with an auto-shutdown.
- Device's output
    - Power level
        - Adjusted by user input (*LEFT/RIGHT* buttons).
        - Affects the electrode's output.
    - Frequency level
        - Adjusted by user input (*Frequency* menu options).
        - Affects the electrode's output.
- Treatment status
    - Sensor detects on-/off-skin status.
    - On-skin status affects electrode's output.
    - Duration of treatment (on-skin).

Figure 19. Simulator Features and Test Cases



Summary

This report presents use cases, an object-oriented analysis, object-oriented design, and traceability matrix for a simulator of software embedded in an MCT device. The goal of the simulator is to allow a designer to assess possible issues arising from the use of such software and its device.

References

Ambler, Scott. 2003-2018. Agile Modeling. www.agilemodeling.com. Accessed February 22, 2020.

Cockburn, A. 2000. “Writing Effective Use Cases”. Pre-publication draft.

Corriveau, Jean-Pierre. [2020]. COMP3004 Course Material. Carleton University. Winter.

Woollaston, Victoria. 2014. “'Star Trek' device that heals skin instantly could soon be used by astronauts on ISS.” DailyMail.com. February 10. Accessed March 31, 2020. https://www.dailymail.co.uk/sciencetech/article-2555731/Star-Trek-device-heals-skin-instantly-soon-used-astronauts-ISS.html

Radonjic, Vojislav. 2020. COMP3004 Course Material. Carleton University. Winter.