| ID | Requirement | Related Use Case | Fulfilled By | Test | Description |
|---|---|---|---|---|---|
| 1 | Application ui contains all required buttons and displays | N/A | MainWindow.ui | Run in Qt and observe | Using Qt's built-in ui framework, the Oasis pro ui is replicated and contains the necessary pushable buttons and is displayed alongside an admin tab to allow for simulation of certain events |
| 2 | Application battery is dependent on the length of therapy, intensity, and connection | N/A | Device.cpp<br>Battery.cpp | Begin a session and observe the progress bar in the admin tab or the console output | The *Device* class holds an instance of the *Battery* class. When a session begins, the battery will constantly be draining, which is handled by the drainBattery() function within the device class. Depending on the length of the session, the battery will be drained accordingly from the startSession function. The higher the intensity, the higher the value to drain the battery will be. Battery will drain faster or slower depending on connection as well. |
| 3 | User can view previous saved session settings | Loading a Recorded Session (UC-08) | MainWindow<br>Device<br>HistoryManager | While a session is not in progress, click the save button and all the previous saved sessions from history manager should be available | The device holds an instance of History Manager. So when the save icon is pressed (curSession is null), the previous sessions that the user saved are visible from the userData.txt file or historymanager instance. The loadSession function is called when the user has selected the session by using the arrow icons and will load that session by chaing the icons for type, length, and intensity. |
| 4 | The user can store their current session settings for later use | Recording a Session (UC-07) | MainWindow<br>Device<br>HistoryManager | When a current session is in progress and the user wants to save that session, the save icon must be pressed and it should be | There is a saveSession function from within the device class that handles saving a function by calling the history managers saveSession function and passing in the current session. The saveSession function within history manager class pushes the current session into the sessions |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  |  | added to the visible table widget and history manager. | vector data member and adds it to the ui. |
| 5 | Application supports 4 Session types | N/A | Session<br>SessionType<br>Device | Press the power button and use arrow icons to choose a type | SessionType class holds the implementation for a session type that requires minimumHerz, MaximumHerz, one CESType and a display icon or two. This class is extensively used by device when creating or modifying instances of session types. |
| 6 | Application supports 3 session lengths | N/A | Session<br>SessionLength<br>Device | Press power button to cycle through session lengths and press check icon once confirmed. | SessionLength class contains the implementation for session length that requires a duration (int), and a customPass (bool). The 3 different session lengths can be altered by the device class when the user presses the power button. Device also initializes several session lengths. |
| 7 | Application supports 8 intensity levels | N/A | Session<br>Device | After selecting session length and type, use arrow keys to select an intensity on the graph from 1-8. | Intensity is a value pertaining within the session class that can be incremented or decremented depending on what the user prefers from the device class. There are 8 different intensity levels and any attempt to go below 1 or above 8 will not work. |
| 8 | The user can select 1 of 4 session Types | Select Session Type (UC-03) | Session<br>SessionType<br>Device<br>definitions | Press the power button and use arrow icons to choose a type | The device class handles the event where the user selects the up or down arrows to change the intensity, where the device would call the sessions' respective increment or decrement intensity functions. |
| 9 | The user can select 1 of 3 session lengths | Select Session Length (UC-02) | Session<br>SessionLength<br>Device<br>definitions | Press power button to cycle through session lengths and press check icon once | If the user is in the process of selecting a session length, the device class will set the respective use case and cycle through the 3 different lengths by handling when the power button gets clicked by the user. The |

| | | | | | |
|---|---|---|---|---|---|
| | | | | confirmed. | device class also handles the consecutive case when the user selects the check icon to confirm the session length. |
| 10 | The user can modify the intensity levels during a session | Adjust Intensity (UC-05) | Session<br>Device<br>definitions | After selecting session length and type, use arrow keys to select an intensity on the graph from 1-8. | The current session attribute within the device class should be able to retrieve the current intensity through a getter. Then that attribute can be modified when the device sets the use case to adjusting intensity and handles the event where the user selects the arrows to increment or decrement the current intensity. |
| 11 | A saved session saves the Type, Length, and intensity | Recording a session (UC-07) | Device<br>HistoryManager<br>Session<br>SessionType<br>SessionLength<br>definitions | After all session attributes have been selected, press the save icon to save session to history manager | Device class contains an instance of a history manager which can take in sessions. Device sets the appropriate use case and handles the event where the save icon is pressed when the current session is available and can be appended to history manager which should be visually displayed and written to a file called userData.txt. |
| 12 | Application can be turned on and off. When off, normal functionality is disabled | Turn on device | Device<br>Battery<br>DisplayIcon<br>MainWindow<br>definitions | To turn on the device, the user must simply press and hold the power icon. To turn off the device, hit the power icon and wait for lights and graph to dim. | The mainwindow is what initializes the user interface and the device class contains a function named turnOn() that illuminates all the icons, sets the table widget, sets the respective use case and starts the timers corresponding to the device battery life. There is a similar turnOff() function within device that stops the timers, ends session, dims icons, turns off the battery and sets the usecase to blank. Both these functions are handled depending on the users actions and utilizes implementation from battery, displayicon and initially mainwindow. |
| 13 | At the start of a session a | N/A | Device | To start the | A function residing within the |

| | | | | | |
|---|---|---|---|---|---|
| | timer begins to run the session for the designated length | | Session<br>SessionLength | timer, begin by selecting all attributes for a session including length, type and intensity. | device class called start function is called in the loadSession function and the handleCheck function in order to begin the current session. This function handles the battery timer as well as the session timer for the designated length chosen by the user. |
| 14 | The session stops if an adequate connection is not established | N/A | Device<br>definitions | Running the user interface without selecting the connect ear clips will ensure that adequate connection is not established and no sessions will ever start. | Within the definitions class, the connection can either be {none, okay, excellent}. In the device class, the startSession function always calls the testConnection function at the beginning which will pause the session if theres no connection. Proceeding the call to testConnection if the connection is still none, then the output will display that the session can't be started because of no connection and the function returns early. |
| 15 | The battery level is displayed via the graph periodically while a session is running and for 5 seconds when the device is turned on | N/A | Device<br>Battery<br>Session | Run session or power on device by hitting the power icon and observe. | The device handles the battery timer where the battery level is shown to the user visually by the graph. Occurs either when the device powers on for the first time or while a session is running. The battery timer itself is used with a signal and slot and is altered within startSession() and turnOn() functions. |
| 16 | The connection is checked when a session is selected and the connection level is displayed via the graph for 3 seconds. | N/A | Device<br>Session<br>definitions | After selecting the final attribute of a session (intensity) and selecting the check icon, the connection is tested and displayed. | Initially there is no connection, however within device there is a testConnection function that tests if the connection is of any of the values from definitions.h (none,okay,excellent) as well as if the correct use case is active. Either the session will be paused or unpaused and if the connection is made, it should be displayed on the graph using a displayConnection() function. The displayConnection function primarily focuses on illuminating |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | the appropriate graph icons. |
| 17 | At the start of the session, unless previous session settings are loaded, the intensity level begins at 1 and must be raised by the user | Adjust Intensity (?) | Device<br>Session<br>DisplayIcon definitions | After selecting session length and type, use arrow keys to select an intensity on the graph from 1-8. Once the session begins, the intensity can be further adjusted by the arrow icons. | Session contains the data member of current intensity that can be altered by the device class when it needs to handle the event when the user clicks the up or down arrows. In which case the increment or decrement functions within the session class are called from device. The handledownArrow and handleUpArrow functions, change the sessiontype or intensity depending on the current use case. |
| 18 | When a session finishes, the device enters into a soft off state | Ending a session | Device<br>Session<br>DisplayIcon definitions | Either if the session ends by the user or the timer, the power icon should be pressed. This will dim the graph icons one by one, device will enter soft off state and device will power off. | The end session function within the device class stops the session timer, dims the respective display icons of the ui, and finally sets the currentSession and curUseCase to blank or nullptr. The softoff bool value from the session class would be altered by using the current session data member from device. |
| 19 | When the battery drops below 15% a warning is displayed to the user | Handling critically low battery | Device<br>Battery definitions | Leave device on until battery is below 15%, the device will let the user know from the console to charge the battery. If this isn’t handled - handling critically low battery use case becomes active. | The device class mainly handles the battery at a critical level, where it outputs to the user that there’s low battery and if it reaches 0, turns off the device and if a session is running, it will end that session. In either case there will be no use case so it’ll be set to blank and finally call the displayBatteryLevel function which simply resets the graph before and after illuminating and flashing the corresponding icons. |