

Traceability Matrix

ID	Requirement	Related Use Case	Fulfilled By	Test	Description
1	The application interface contains buttons, display, and electrodes.	N/A	MainWindow.ui	Run the simulator in Qt to observe the ui.	Using QT's built in user interface framework, the physical DENAS system was replicated. Also, all buttons are clickable with the mouse. The DENAS is displayed alongside an admin panel, which allows for more precise control of specific elements of the simulator system.
2	The application battery level is dependent on time and power level of therapy.	N/A	MainWindow, Therapy, Profile	Start a treatment, and observe the battery icon: battery level will lower as the treatment progresses.	The <i>Profile</i> class keeps a consistent record of the power level of the device, which the <i>MainWindow</i> can then use to update its display. As a therapy is run, the battery level will consistently decrease (rate of decrease is affected by the power level). The user's profile will store this information, and the <i>MainWindow</i> will update its information.
3	Treatment displays the time of treatment.	N/A	MainWindow, Record, Therapy	Select a treatment to observe its allotted time.	<i>Therapy</i> classes contain a <i>QTimer</i> attribute, which will maintain an internal time of how long the therapy has been running. The <i>MainWindow</i> will display this information when a therapy is running, and a <i>Record</i> class will be created to save it when it ends.
4	Treatment time advanced only when on skin (References Requirement 3).	Activate Electrodes Use Case (UC6)	MainWindow, Record, Therapy	Start a treatment, and observe the timer when toggling the <i>apply to skin</i> button.	The <i>MainWindow</i> class keeps track of whether or not the device is pressed to the skin (on skin status can be changed in the admin panel or via the small button at the bottom of the device image). The therapy timer will

					only advance if the device is considered to be on skin.
5	Treatment time pauses (or does not advance) when not applied to skin (References Requirement 3).	Activate Electrodes Use Case (UC6)	MainWindow, Record	Start a treatment, and observe the timer when toggling the <i>apply to skin</i> button.	Implementation here is the opposite of the previous case. The timer will not advance if the device is considered to be off skin.
6	Device supports a power level of a therapy in the range [0, 100].	Change Power Level Use Case (UC3)	MainWindow, Record, Profile	Select a treatment, and change power level by clicking either the <i>left</i> or <i>right</i> button or modify the power level in the admin panel.	Power level can be changed via the left and right buttons on the denas simulator when a therapy is running, or by the power counter in the admin panel to the right. When any change in the power level is detected, the <i>Profile</i> class will store the information and update the <i>MainWindow</i> .
7	Electrodes are simulated for application on skin.	Activate Electrodes Use Case (UC6)	MainWindow	Start a treatment, and toggle the <i>apply to skin</i> button, simulating electrode application.	The small silver button on the bottom of the DENAS simulator will simulate pressing the device to the skin when pressed (the state can also be changed via the dropdown in the admin window). When this happens, the rest of the <i>MainWindow</i> class and the display will be updated, and any timers will be started/stopped.
8	Device supports 4 frequencies.	User Selects a Frequency (UC5)	MainWindow, Menu, Therapy	Select the <i>Frequencies</i> menu to view device's supported frequencies.	The Frequencies are represented as <i>Therapy</i> objects. These have a name, timer, operating frequency, and max time (not implemented for frequency mode). The <i>MainWindow</i> stores these objects, as well as the <i>Menu</i> objects that correspond to them.
9	Device supports 4 programs.	User Selects a Program (UC4)	MainWindow, Menu, Therapy	Select the <i>Programs</i> menu to view device's supported programs.	The Programs are represented as <i>Therapy</i> objects. These have a name, timer, operating frequency, and max

					time. The <i>MainWindow</i> stores these objects, as well as the <i>Menu</i> objects that correspond to them.
10	The user can choose to record a therapy and add to history of treatment for a single user.	Save a Therapy Use Case (UC9)	Record, DBManager	Start a treatment, and press either the <i>back</i> , <i>menu</i> , <i>power</i> button, or complete a program to save to the history of treatments.	All therapies are recorded automatically whenever a treatment finishes or is closed prematurely. Their information (start time, duration, etc) is stored in a corresponding record object and then saved to an external database via the <i>DBManager</i> object in <i>MainWindow</i> .
11	The user can view a history of treatment.	View Saved Therapies Use Case (UC7)	MainWindow, Menu, Record, DBManager	From the <i>History</i> menu, select the <i>View</i> option to view all treatments currently saved to the device.	Through using the buttons on the DENAS simulator, the user can navigate to the “HISTORY” and “VIEW” menus and see a scrollable list of all the therapies the device has run since they were last cleared. These records are loaded from the database via <i>DBManager</i> into <i>MainWindow</i> .
12	The user can clear their history of treatment.	Clear Saved Therapies Use Case (UC8)	MainWindow, Menu, DBManager	From the <i>History</i> menu, select the <i>Clear</i> option to clear all treatments currently saved to the device.	Through using the buttons on the DENAS simulator, the user can navigate to the “HISTORY” and “CLEAR” menus and choose to clear the data from the system. If “NO” is selected, then they will be taken back to the previous menu with no changes. If “YES” is selected, then the <i>MainWindow</i> will instruct the <i>DBManager</i> to clear the database, and the users will be taken back to the previous menu.
13	The application does not contain any memory leaks.	N/A	N/A	Run valgrind to check for memory leaks.	All dynamically allocated memory that the program was designed to allocate is deleted in the appropriate class destructor.

14	The device simulation can be turned on and off, disabling normal device functionality when the device is turned off.	Turn Device On Use Case (UC1), Turn Device Off Use Case (UC2)	MainWindow	Turn the device on/off by toggling the <i>power</i> button. Select other buttons while the device is off, and observe that their functionality has been disabled.	The power button, when clicked, sends a signal to <i>MainWindow</i> to disable/enable the DENAS screen and all the buttons on the DENAS simulator.
15	Power level of the device is only changeable during a treatment.	Change Power Level Use Case (UC3)	MainWindow	Select a treatment, and change power level by clicking either the <i>left</i> or <i>right</i> button. Next, try changing power level using the same method on the main menu; power level will not change.	When outside of a treatment, <i>MainWindow</i> blocks all signals to the left and right buttons as they are not currently in use. When a treatment is running the signals are reenabled.
16	A saved record saves the date, time, type of treatment, highest power level, and the duration of the treatment.	Save A Therapy Use Case (UC9)	MainWindow, Record, DBManager	N/A	The <i>Record</i> class has attributes for the date, time, type, highest power, and duration of treatments. The <i>MainWindow</i> class keeps a collection of <i>Record</i> objects as they are created. The <i>DBManager</i> will store and retrieve older records from other runs of the simulator as is required.
17	The device becomes non-functional when the battery level reaches 0.	Turn Device On Use Case (UC1)	MainWindow, Profile	Drain the battery level by running a treatment. Once the battery level reaches 0 and the device turns off, try selecting any buttons on the device, and observe that their functionality has been disabled.	When the battery level reaches 0, the function that handles battery reduction alerts the rest of the <i>MainWindow</i> class to turn off the DENAS simulator. If the <i>power on</i> button is pressed while the battery is at zero, the simulator will not turn on.
18	At the start of a frequency treatment, the timer starts	N/A	MainWindow, Therapy	Start a frequency treatment, and observe the timer.	The <i>Therapy</i> classes maintain a <i>QTimer</i> attribute and a max time. For Frequencies, the max time is

	at 0:00 and counts up (Requires 4).				disregarded and the timer starts at zero, counting up instead of down.
19	At the start of a program treatment, the timer starts at the time recommended for the particular program, and counts down to 0:00 (Requires 4).	N/A	MainWindow, Therapy	Start a program treatment, and observe the timer.	The <i>Therapy</i> classes maintain a <i>QTimer</i> attribute and a max time. For Programs, the visual timer will start at the max time and subtract time off of it until it reaches zero, at which point the treatment ends.
20	Records are stored in persistent storage.	N/A	DBManager	Have record states observable in the <i>View</i> menu, and restart the application.	<p>The <i>DBManager</i> class is used to create tables, insert/delete, and query data in an external SQLite database.</p> <p>The <i>DBManager</i> will insert <i>Record</i> objects created by <i>MainWindow</i> into the database. It will also retrieve the created records later for the <i>MainWindow</i> class to hold.</p>
21	Treatment information is stored in persistent storage.	N/A	DBManager	N/A	<p>The <i>DBManager</i> class is used to create tables, insert/delete, and query data in an external SQLite database.</p> <p>The <i>DBManager</i> creates the information that will be held by <i>Therapy</i> objects and stores it in the database for easy retrieval.</p>
22	Last battery level is stored in persistent storage.	N/A	DBManager	Set or observe a change in battery level, and restart the application.	<p>The <i>DBManager</i> class is used to create tables, insert/delete, and query data in an external SQLite database.</p> <p>The <i>DBManager</i> saves a record of a user profile, which is loaded when the simulator starts and updated when the simulator is closed. This profile keeps a record of the battery level,</p>

					allowing it to persist between runs.
23	Last power level is stored in persistent storage.	N'A	DBManager	Set or observe a change in power level, and restart the application.	<p>The <i>DBManager</i> class is used to create tables, insert/delete, and query data in an external SQLite database.</p> <p>The <i>DBManager</i> saves a record of a user profile, which is loaded when the simulator starts and updated when the simulator is closed. This profile keeps a record of the most recent power level, allowing it to persist between runs.</p>