Traceability Matrix

ID	Requirement	Related Use Case	Fulfilled By	Test	Description
1	A light on the machine and/or a symbol on the screen that indicates an active pulse reading	Use Case 3: Initiate and Complete a Session Part 3	LED, MainWindow	Start a session and observe the “Length” metric when toggling the HRV Signal button.	Using the LED class we implemented, the light is togglable with a “HRV Signal'' button. It toggles between on and off depending on button clicks, which simulate sensor activations.
2	A suggested user interface consists of the following main components: A screen and buttons. The screen contains the menu options and the display graph. There are eight buttons: an off/on button for the device, a menu	N / A	MainWindow.ui	Start the simulator in QT and view the application interface	Using the QT user interface framework, we devised a three-part interface, with a left sidebar with power, HRV Signal, and battery widgets; a centred screen which displays session information and menus; and the right sidebar, which

	button, a standard back button which will return the user to the menu, four arrow buttons (up/down,left/right) and a selector in the center of the arrow buttons which also functions as a start/stop button in session mode.				contains the navigational interface.
3	In addition, the device has a led light that changes to red, blue or green to indicate coherence level	Use Case 3: Initiate and Complete a Session Part 5	LED, MainWindow, Session, Reader	Start a session and observe the coherence level light in the upper centre of the interface.	The LED class packages the turnOn(), turnOff(), and setColour() functions together. The LED that is on during the Session is part of the MainWindow class. It uses the Session class as an intermediary to

					receive information from the Reader class about when to change colour and what to. Off and on is handled by MainWindow.
4	Press selector to initiate and end a session. The menu options are displayed as default on the session screen. There is an option at the top to start a new session. The menu could consist of the following options: start new session, settings, log/history	N / A	Menu, MainWindow	Start the simulator and observe the main menu. Use the select button to start a session, and use the select button to then end the session.	The Menu class controls menu indexing functionality and consists of a parent-child structure with Menu QVectors to store children. The main menu consists of Start Session, Logs/History, and Settings. When a session is started the selector gains the ability to end the session.

5	Session screen must display the main HRV graph (HR vs time) with key metrics	Use Case 3: Initiate and Complete a Session Part 4	MainWindow, Session, Reader, QCustomPlot	Start a session and view the active session screen.	The Session class handles all session related information, and stores the key metrics in its member variables. The plotDataPoints() function in MainWindow handles the session information. It receives information from Session getter functions, which in turn receive data from the Reader class, which generates data from file. plotDataPoints() also handles the graph using Reader data with the QCustomPlot class.
6	The metrics on the	Use Case 3: Initiate	Session, Reader,	Start a session and	The Session class

	screen include the current coherence score (numerical value), length (duration of session), achievement (total sum of coherence scores sampled every 5 seconds)	and Complete a Session Part 4	QCustomPlot	view the active session screen, noting the labels above the graph and their change over time.	handles all session related information, and stores the key metrics in its member variables. The Reader class consists of a group of functions which are used to generate and calculate metrics, so that they can be passed to Session. The Reader class also uses the QCustomPlot class to plot the graph.
7	A breath pacer in the form of a strip of lights on the machine itself, or a ball going back and forth on the session screen, default set at one breath every 10	Use Case 3: Initiate and Complete a Session Part 6	LightStrip, MainWindow	Start a session and view the active session screen and the lightstrip below the graph.	The LightStrip class uses a double pointer of LED objects to form the strip, and uses a QTimer object to handle the on and off of the lights. The MainWindow class

	seconds, adjustable in settings				uses a LightStrip as the Breath Pacer and passes in the default value of 10 seconds for the setInterval() function. This can be changed through the Breath Pacer Settings.
8	The settings tab includes challenge level (optional) and breath pacer settings	Use Case 6: Modifying Breath Pacer Settings & Use Case 7: Modifying Challenge Level	Menu, MainWindow	Start the simulator and navigate to the settings tab using the directional and select buttons. Select the settings tab and view the sub-settings options.	The Menu class controls menu indexing functionality and consists of a parent-child structure with Menu QVectors to store children. The Settings tab is the parent to the Challenge Level and Breath Pacer settings tabs, which open onto new pages where their

					values can be modified.
9	(optional) There are 4 challenge levels for coherence, from beginner to advanced, for the user to choose	Use Case 7: Modifying Challenge Level Part 6	Session, Reader	Select the settings tab and then select Challenge Level.	The Session class handles all session related information, and stores the key metrics in its member variables. Challenge level is passed to the Session class through the constructor and the Reader class uses the value to specify how coherence level is determined.
10	The breath pacer, 1-30 seconds, increases time interval between each breath, default at 10 seconds	Use Case 3: Initiate and Complete a Session Part 7	LightStrip, LED, MainWindow,	Start a session and observe the breath pacer lightstrip below the graph, over the course of the session.	The LightStrip class uses a double pointer of LED objects to form the strip, and uses a QTimer object to handle the on and off of the lights.

					The MainWindow class uses the setInterval() function to ensure that whatever the Breath Pacer interval value, the breaths are synchronized.
11	When the user ends a session a summary view will appear that includes the following information: challenge level (optional), percentage of time in different coherence levels (low, medium and high), average coherence, length of session, achievement score, entire HRV graph	Use Case 3: Initiate and Complete a Session Part 11	Session, MainWindow	Start a session and then use the centre selector to end it. View the resulting session summary screen.	The Session class handles all session related information, and stores the key metrics in its member variables. The session summary view is handled by the MainWindow user interface which displays all information along with the complete graph.

12	The menu contains a log or history tab of all sessions, with dates, when selected show the summary view, as well as the ability to delete a session	Use Case 4: Viewing Logged Session Part 5 & Use Case 5: Deleting Logged Session Part 6	Menu, Session, MainWindow	Start a session and then use the centre selector to end it. Use the centre selector to then return to menu, and then select the Logs/History tab. Select the most recent session and view its information, and use the delete button below to remove the session.	The Menu class controls menu indexing functionality and consists of a parent-child structure with Menu QVectors to store children. The Session QVector in the MainWindow class stores all Session logs. These logs are used by the Menu class to create menu options for each one, and the Session objects are displayed by MainWindow when opened, similar to their summary view. As they are stored in the QVector, they

					can be easily deleted with their index.
13	An option to reset, wipe all data and restore the device to the initial install condition	Use Case 8: Resetting Device Part 4-8	MainWindow, Menu	Start a session and then use the centre selector to end it. Use the centre selector to then return to menu, and use the directional buttons to navigate to and select the Settings tab. Use the down button and select button to select Reset Device. Confirm that you want to reset the device by clicking select. Now observe the Logs/History tab.	Logs/History for the previous sessions are stored in a Session vector in the MainWindow class. Settings for Challenge Level are stored in a member variable and for the Breath Pacer are stored in the input slider itself. Using the Reset Device menu option resets the menus, clears the Session vector, and resets the Settings values.
14	There is a battery charge indicator on the session screen	N / A	Battery, MainWindow	Start the simulator and view the battery slider on the left side of the	The Battery class implements a QSlider widget which continuously

				interface.	decreases downwards and displays the percentage. It can be moved upwards to recharge.
15	A beep goes off when a new coherence level is reached	Use Case 3: Initiate and Complete a Session Part 8	MainWindow, Session, Reader	Start a session and observe the console output as the graph changes.	The MainWindow class has a QDebug statement which outputs the “Beep” upon reaching a new coherence level, which is determined by the Session and Reader classes.