

Traceability Matrix:

| ID | Requirement | Related Use Case | Fulfilled By | Test | Description |
|----|--|------------------|------------------------------------|---|--|
| 1 | The application interface consists of a screen which contains the menu options and the display graph, and eight buttons. | N/A | MainWindow.ui | Run the application in Qt to observe the UI elements and how it responds to user requests. | The HeartWave uses Qt's built-in user interface for users to interact with the app. The GUI includes user UI which is composed of a screen and 8 buttons. The screen shows main menu and submenus, among which "START NEW SESSION" page and "VIEW" page display HRV graph. User can click each button to interact with the app. Each button is connected to a specific slot. For example, power button is connected to the powerSwitch slot. Additionally, there is a control panel UI for battery level setting and battery charging. |
| 2 | A symbol on the screen that indicates an active pulse reading. | UC2 | MainWindow, MainWindow.ui | From the main menu, select "START NEW SESSION" option, and observe whether the symbol is displayed or not together with updated HRV graph after clicking sensor button. | There is HR contact symbol displayed on the top right of the screen during session measurement, indicating that sensor is connected and the pulse measurement is ongoing. The activeSensor() function in MainWindow class checks if the sensor is connected or not and the symbol is set visible in line with the sensor's state. |
| 3 | The device has a led light that changes to red, blue | UC2 | MainWindow, Session, MainWindow.ui | From the main menu, select "START NEW SESSION" option and press sensor button, | There are three labels located above the screen, which will be displayed when a corresponding coherence level reaches |

| | | | | | |
|---|--|----------|---------------------------------|--|---|
| | or green to indicate coherence level. | | | then check the presence of three labels where red label represents low coherence level, blue label represents medium coherence level, and green label represents high coherence level. | during session measurement. The red label represents low coherence level. The blue label indicates medium coherence level. The green label is for high coherence level. updateSessionView() function in MainWindow class is responsible for displaying the appropriate coherence level label (low, medium, or high) according to calculated coherence score. |
| 4 | The menu options are displayed as default on the session screen. The menu consists of the following options: start new session(at the top), settings, history. | UC1, UC2 | MainWindow, Menu, MainWindow.ui | Start the simulation and press the power button, then check if it displays main menu and the three options | The MainWindow class contains menu objects and methods of setupConnections() and initializeMenu() to create and initialize the main menu tree and the main menu view. The Menu class is a recursive structure for creating menu and submenus. |
| 5 | Press selector to initiate and end a session. | UC2 | MainWindow, Session | From the main menu, press select button to select the "START NEW SESSION", check if the screen changes to session view. After starting the session measurement, press select button to observe if the session ends and saves data to database. | The MainWindow class contains selectAction() function that handles the selector button action. The method checks different scenarios. When user chooses the "START NEW SESSION" option on the main menu, if the battery level is above 15%, the startSession() function will be called and a new session object will be created. When user presses the button after the measurement starts, endSession() will be called, which ends the current session and saves the data to database and record object. |
| 6 | Session screen must | UC2 | MainWindow, | From the main menu, select | The Mainwindow class contains plot() |

| | | | | | |
|---|--|-----|--|---|--|
| | display the main HRV graph (HR vs time) with key metrics. | | Session, QCustomPlot | “START NEW SESSION” option and press sensor button, then check if the HRV graph and metrics such as coherence score, session length, and achievement score are displayed and updated. | method, which uses QCustomPlot class's methods to plot the HRV graph and adds the graph to sessionGraph widget during a session measurement. The X-Axis represents time and Y-Axis represents heart rate. The maximum length of a session is set to 120 seconds. The simulateHeartIntervals() function in Session class generates random heartbeat intervals as the original data for the program. The HRV graph updates every 5 seconds. |
| 7 | On the device there should be a light that changes to red, blue or green indicating low, medium or high coherence, depending on the challenge level. | UC2 | MainWindow, Session, Setting MainWindow.ui | From the main menu, select “START NEW SESSION” option and press sensor button, then check the presence of three labels according to different challenge levels. To change challenge level, from the main menu, select “SETTINGS” option and press down button to move to challenge level option. Then use left or right button to decrease or increase the parameter. | There are three labels located above the screen, which will be displayed when a corresponding coherence level reaches during session measurement. The red label represents low coherence level. The blue label indicates medium coherence level. The green label is for high coherence level. The Session class has updateCoherenceLevel() method that update the coherence level according to thresholds of 4 challenge levels. updateSessionView() function in MainWindow class is responsible for displaying the appropriate coherence level label (low, medium, or high) according to calculated coherence score and thresholds of different challenge levels. |
| 8 | The metrics on the screen include the current | UC2 | MainWindow, Session | From the main menu, select “START NEW SESSION” | The Session class has corresponding attributes such as coherenceScore, length, |

| | | | | | |
|---|--|----------|------------------------------|---|--|
| | coherence score (numerical value), length (duration of session), achievement (total sum of coherence scores sampled every 5 seconds). | | | option and press sensor button, then check if the metrics such as coherence score and achievement score are displayed and updated every 5 seconds. Check if session length is displayed updated every second. | achievementScore and QTimer timer. sessionTimerSlot() as a slot function in MainWindow class is responsible for updating session. Length(displayed by lengthValue QLabel) is updated every second while coherenceScore (displayed by coherenceValue QLabel) and achievementScore(displayed by achievementScore QLabel) are updated every 5 seconds. |
| 9 | 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 seconds, adjustable in settings. | UC2, UC5 | MainWindow, Session, Setting | From the main menu, select “START NEW SESSION” option and press sensor button. Once the measurement start, check the presence of the breath pacer progress bar. To change breath pacer interval, from the main menu, select “SETTINGS” option and press down button to move to breath pacer interval option. Then use left or right button to decrease or increase the parameter. | The breath pacer is represented by a progress bar at the bottom of the screen, going forth and back during the session measurement. MainWindow class contains several methods to achieve this functionality. The updateBP() function updates the breath pacer progress based on the given interval. The sessionTimerSlot() function and Session class’s QTimer are used to update breath pacer displaying on screen every second. The initializeMenu() function sets the default value of breath pacer interval to 10. parameterPlus() and parameterMinus() are called when incrementing and decrementing breath pacer interval in the settings menu , which use getters and setters functions of Setting class to get and set the two values. User presses right/left buttons to trigger these two functions. |

| | | | | | |
|----|--|----------|---------------------------|--|--|
| 10 | The settings tab includes challenge level (optional) and breath pacer settings. | UC4, UC5 | MainWindow, Menu, Setting | From the main menu, select “SETTINGS” option by pressing the select button, then press down button to check if there are options for challenge level and breath pacer interval. Test if left or right button can be used to decrease or increase the parameters. | MainWindow class contains several methods to achieve this functionality. The initializeMenu() function creates submenu options for setting menu, which includes "BREATH PACER INTERVAL" and "CHALLENGE LEVEL". parameterPlus() and parameterMinus() functions are called when incrementing and decrementing breath pacer interval or challenge level in the settings menu, which use getters and setters functions of Setting class to get and set the two values. User presses right/left buttons to trigger these two functions. |
| 11 | There are 4 challenge levels for coherence, from beginner to advanced, for the user to choose. | UC4 | MainWindow, Setting | From the main menu, select “SETTINGS” option by pressing the select button, then press down button to check if there is an option for challenge level. Test if left or right button can be used to decrease or increase the challenge level and if the range is between 1 and 4. | The initializeMenu() function in MainWindow class creates submenu options for setting menu, which includes "CHALLENGE LEVEL". The default value is 1. MainWindow class also contains parameterPlus() and parameterMinus() functions that are called when incrementing and decrementing the challenge level in the settings menu, which use getters and setters functions of Setting class to get and set. User presses right/left buttons to trigger these two functions. |
| 12 | The breath pacer, 1-30 seconds, increases time interval between each | UC5 | MainWindow, Setting | From the main menu, select “SETTINGS” option by pressing the select button, then press down button to check if | MainWindow class contains several methods to achieve this functionality. The updateBP() function updates the breath pacer progress based on the given interval. The |

| | | | | | |
|----|--|-----|--|---|--|
| | breath, default at 10 seconds. | | | there is an option for breath pacer and if the default value is 10. Press left or right button to check the functionality of decreasing or increasing the breath pacer interval. Check if the range is between 1 and 30. | initializeMenu() function sets the default value of breath pacer interval to 10. parameterPlus() and parameterMinus() are called when incrementing and decrementing breath pacer interval in the settings menu. The range of breath pacer interval is between 1 and 30. |
| 13 | 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. | UC2 | MainWindow, Session, DBController, Record, QCustomPlot | After the session measurement starts, press select/back/mainmenu buttons or wait until the session reaches the maximum session duration(2 minutes) to check if the session ends and displays the summary review with required indicators and HRV graph. | MainWindow class contains several methods to achieve this functionality. endSession() function ends the current session, stops the session timer, saves the session data to DBController class 's database and Record object. displayReview() function displays a summary review which includes HRV graph and all required indicators. plotHistory() method uses QCustomPlot class's methods to plot the HRV graph and adds the graph to historyGraph widget . |
| 14 | 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. | UC3 | MainWindow, Menu, DBController, Record | From the main menu, check the presence of History tab. Select the option and observe if a list of records is shown. Then press selector button to enter a record's submenus. Choose "VIEW" to check if the correct summary view is displayed on screen. Or choose "DELETE" option to check if it the selected | MainWindow class contains several methods to achieve this functionality. initializeMenu() function creates submenu options for History menu, using date time as the submenu names. It then adds view and delete submenus under each record menu. selectAction() function handles the selector button action, which checks different scenarios. When user chooses "VIEW" option on the Settings page, viewAction() |

| | | | | | |
|----|--|-----|---|--|--|
| | | | | record is deleted or not. | function will be called, which display a record summary based on the date. When user chooses "DELETE" option on the Settings page, deleteAction() function will be called to delete the record from DBController's database and the record object itself, whose start time matches the submenu name. |
| 15 | Device user can wipe all data and restore the device to the initial install condition. | UC6 | MainWindow, DBController, Record, Menu, Setting | Complete at least one new session measurement and change the values of breath pacer interval and challenge level. Then select "RESET" option on "SETTINGS" menu page. Confirm with "YES" option and check if all data has been wiped and if the device has been to set to initial condition. | The initializeMenu() function in MainWindow class creates submenu options for setting menu, which includes "RESET" option. When the option is selected, user will be asked to confirm the reset action. If "YES" option is chosen, then selectAction() function will handle the case to set the device to initial state. |
| 16 | There is a battery charge indicator on the session screen. | N/A | MainWindow, MainWindow.ui | There is a battery setting option on control panel UI. Click spin box to choose desired values to test if the battery color changes to red and if the console prints out a message to remind user when battery capacity is less than 15%. | changeBatteryCapacity() function in MainWindow class changes the color of battery indicator based on current battery capacity. When it is red, which is the case that battery level is below 15%, user will be asked to charge the device. |
| 17 | A beep goes off when a new coherence level is | UC2 | Session MainWindow | From the main menu, select "START NEW SESSION" | beep() function in Session class will emit a beep signal when new coherence level is |

| | | | | | |
|----|----------|------------------------------|---------------------|--|---|
| | reached. | | | option and press sensor button, check if the console prints out “Beep” message when a new coherence level is reached. | reached. updateCoherenceLevel() function is responsible for updating the coherence level of the session during the session measurement. |
| 18 | Sensor | UC2 | MainWindow, Session | From the main menu, select “START NEW SESSION” option and press sensor button, check if the session starts to display data. During an active session, press sensor button to check if the session stops and press sensor button again to observe if the paused session continues or not. | There is a sensor button at the bottom left of the user UI, which is connected to activeSensor() function in MainWindow class to toggle sensor activation. MainWindow class also has sensorOn attribute to indicate if the sensor is connected to the device or not, which is in line with HR contact symbol. The sensor state starts or stops QTimer in Session class. For example, when sensorOn is false, the timer stops. While the timer resumes and the session continues if sensorOn is true. |
| 19 | Battery | UC1, UC2, UC3, UC4, UC5, UC6 | MainWindow, | There is a battery setting option on control panel UI. Click spin box to choose desired values. For example, set the value to 0 to check if the device has turned off. Or set the battery below 15% to check if user can enter “START NEW SESSION” submenu. Press the charge button to check if the battery has been set to full capacity or not. When interacting with the | There is a control panel UI for battery level setting and battery charging. MainWindow class contains several methods to manage battery. changeBatteryCapacity() is the slot to change the battery level and chargeBattery() function charges the battery to full capacity when user presses batteryPushButton. consumeBattery() decreases the battery capacity based on start-new-session case and all-others case. When battery capacity is 0, the device will automatically shut down. When battery capacity is below 15%, user cannot start a |

| | | | | | |
|--|--|--|--|--|---|
| | | | | application, observe if the battery capacity decreases or not. | new session and will be asked to charge the device. |
|--|--|--|--|--|---|