

ID	Requirements	Related Use Case	Fulfilled By	Implemented By	Tested By	Description
1	The device can be turned on when pressing the POWER button, while the battery is enough.	Use Case 2: Power ON or OFF the Device	mainwindow.h	mainwindow.h→on_powerButton_clicked()	ui→POWER button→the screen will lighten.	When the battery is still enough, the POWER button should be able to turn on the device.
2	The device can be turned off when it is on and the POWER button is pressed.	Use Case 2: Power ON or OFF the Device	mainwindow.h	mainwindow.h→on_powerButton_clicked()	ui→POWER button→ the screen will turn black.	The POWER button should be able to turn off the device.
3	When the device turns on, the menu is automatically generated.	N/A	mainwindow.h menu.h	mainwindow.h→initializeMainMenu() menu.h→Menu(), addChildMenu()	ui→POWER button→the menu is shown	There are multiple levels of menus, allowing the user to choose to start a new session, set some settings, and view historical logs.
4	The user can use arrow buttons and the SELECT button, BACK button, and MENU button to travel through different menus.	N/A	mainwindow.h	mainwindow.h→on_backButton_clicked(), on_menuButton_clicked(), on_upButton_clicked(), on_selectButton_clicked(), on_downButton_clicked(), updateMenu(), navigateDownMenu(), navigateUpMenu(), navigateSubMenu(), navigateBack(), navigateMainMenu()	ui→UP/DOWN to navigate up and down in the same menu level; ui→BACK to go back to the previous menu; ui→MENU to go back to the main menu	The SELECT button is used to navigate to the submenu, choose setting options, and start a new session. The BACK button is used to go back to the upper-level menu. The MENU button is used to redirect to the main menu from the current directory.
5	The breath pacer can be set by the user inside the setting menu.	Use Case 3: Set the Breath Pacer	mainwindow.h	mainwindow.h→navigateSubMenu(), updateMenu()	ui→SETTING→BREATH PACER→set the breath pacer	The user can set the breath pacer under the BREATH PACER under the SETTING menu.

6	The challenge level can be set by the user inside the setting menu.	Use Case 4: Set the Challenge Level of Coherence	mainwindow.h	mainwindow.h→navigateSubMenu(), updateMenu()	ui→SETTING→CHALLENGE LEVEL→set the challenge level	The user can set the challenge level under the CHALLENGE LEVEL under the SETTING menu.
7	A new session starts when the user selects the specific selection on the menu.	Use Case 5: Start New Session	mainwindow.h	mainwindow.h→navigateSubMenu()	ui→START NEW SESSION→start a new session	The user selects the START NEW SESSION and selects SELECT, and a new session starts.
8	The coherence score, time, and achievement will be displayed on the screen during a session.	Use Case 5: Start New Session	mainwindow.h	mainwindow.h→sessionUI()	ui→during a session, the session information will be displayed at the top of the screen	The screen will display the coherence score, session length, and achievement.
9	The device receives new HRV data during a session every second.	Use Case 5: Start New Session	mainwindow.h hrvtable.h	mainwindow.h→addNextInteger() hrvtable.h→cohGetAt(), incGetAt()	The session info and the HRV graph change every second.	New HRV data from HRVtable class is sent to the MainMenu class every second, to simulate the sensor.
10	The device calculates the coherence score every five seconds.	Use Case 5: Start New Session	mainwindow.h hrvtable.h	mainwindow.h→hrvtable.h→cohScoreGetAt(), incScoreGetAt()	The coherence score changes every five seconds during a session.	A new coherence score data from HRVtable class is sent to the MainMenu class every second, to simulate the coherence calculation.
11	The HRV graph will be displayed on the screen during a session.	Use Case 5: Start New Session	mainwindow.h waveitem.h	mainwindow.h→on_HRVaddButton_clicked() waveitem.h→WaveItem(), paint()	The HRV graph is displayed in the middle of the screen during a session.	The screen will display the real-time HRV graph.

12	The LED lights above the screen will light according to the coherence level.	Use Case 5: Start New Session	mainwindow.h	mainwindow.h→redLED_on(), redLED_off(), blueLED_on(), blueLED_off(), greenLED_on(), greenLED_off()	The LED lights above the screen automatically turn on and off according to the coherence score and the challenge level during a session.	When the coherence score is at the low level, the red LED lightens; when it is at the medium level, the green LED lightens; when it is at the high level, the blue LED lightens.
13	When the user reached a new coherence level, there will be a BEEP sound.	Use Case 5: Start New Session	mainwindow.h	mainwindow.h→redLED_on(), blueLED_on(), greenLED_on()	When the user reached a new coherence level, a BEEP message is displayed.	A BEEP message is displayed every time the user achieves a new coherence level to simulate the BEEP sound.
14	The user can press the SELECT button to end an ongoing session.	Use Case 6: End a Session	mainwindow.h	mainwindow.h→navigateSubMenu()	Press SELECT during a session to end the session, and the summary is displayed.	The user can end a session by pressing SELECT during a session.
15	No other buttons except SELECT and POWER can interrupt an ongoing session.	N/A	mainwindow.h	mainwindow.h→sessionUI()	No other buttons work during a session.	The device will check the current states, if a session is running, signals from other buttons will be ignored.
16	When a session ends, a summary including textual session details as well as the entire HRV graph is displayed.	Use Case 6: End a Session	mainwindow.h waveitem.h	mainwindow.h→sessionUI() waveitem.h→WaveItem()	The entire HRV graph of a session is displayed inside the summary.	The entire HRV graph of the session is displayed in the session summary.
17	The user can finish reviewing the summary by pressing the SELECT button again.	Use Case 6: End a Session	mainwindow.h	mainwindow.h→navigateSubMenu()	Press SELECT when reviewing the summary to go back to the main menu.	The user presses the SELECT button to finish reviewing the summary and goes back to the

						main menu.
18	The summary data will be stored in the device automatically.	Use Case 7: Check, Delete, and Wipeout Historical Data	mainwindow.h	mainwindow.h→recordAndDate mainwindow.h→navigateSubMenu();	press SELECT after viewing summary, data are stored in the variable in MainWindow	When a session end, the data is stored in a QStringList recordAndDate
19	The user can view historical data inside the history menu.	Use Case 7: Check, Delete, and Wipeout Historical Data	mainwindow.h	mainwindow.h→navigateSubMenu()	When the user chooses log/history, specific date, and view. A summary and a graph shows.	When the user select view, data shows with corresponding date
20	The user can delete specific data using the DELETE button.	Use Case 7: Check, Delete, and Wipeout Historical Data	mainwindow.h	mainwindow.h→navigateSubMenu()	When the user chooses log/history, specific date, and delete. It goes back and this record no longer exists.	When the user select delete, select the ith element to delete
21	The user can wipe out all historical data using the RESET button.	Use Case 7: Check, Delete, and Wipeout Historical Data	mainwindow.h	mainwindow.h→navigateSubMenu()	When user chooses log/history and choose reset, user goes log/history again and there is no records	When user select reset, all records removed and get use back to the start menu
22	The battery level will go down every time a session runs.	N/A	mainwindow.h	mainwindow.h→navigateSubMenu()	ui→battery goes down by 10 after every session.	The battery goes down every time a session ends.
23	When the battery ran out, the device will be turned off automatically, and cannot be turned on again until the battery is charged.	N/A	mainwindow.h	mainwindow.h→on_powerButton_clicked()	The screen turns black, and the POWER button does not work if the battery goes to 0.	The device will turn off when the battery ran out, before that the session data is stored.
24	The user can charge the device till full.	Use Case 8: Charge the Device	mainwindow.h	mainwindow.h→on_chargingCheckBox_stateChanged()	ui→charging→click to charge battery.	The device can be charged, and it stops charging automatically when

						it is full. If the user wants to charge the device when it is full already, a message will be displayed.
--	--	--	--	--	--	--