Team Project: Developing and testing a software-based prototype of a Heart Rate Variability device

AUTHORS :

GROUP 13
Braydon Hoard | 101144599

Saajid Aliyar | 101153376

Ron Stuchevsky | 101188412

Maryam khalaf | 101032990

SUBMITTED TO

Prof Vojislav D Radonjic

COMP3004 AB: Object-Oriented Software Engineering
Winter 2023

Carleton University

Table of Contents

Use Case ID: UC1

Name: Normal Operation
Actor: User
Pre-condition: The heatwave device is charged and turned on.
The user already knows the device's Ui and functionality.
post-condition
The new session information is saved and the user is able to continue performing actions on the device.
Complete Scenario:
Normal Success Scenario:
1. User presses the "On" button to turn on the heart wave application.
2. User connects the device sensors to his body.
3. Sensor heart icon on the device turns on indicating stable connection.
4. User moves through the menu using the arrow buttons selecting "Start New Session" by navigating through the selector.
5. The user presses the selector to initiate the session, and the breath pacer on the session screen continues to guide the breathing.
6. The user concentrates on their breathing and the HRV graph is displayed on the screen.
7. The device's LED is lit and indicates their coherence level (red, blue, or green for low, medium, or high coherence, which depends on the challenge level).
8. The user's coherence score, session duration, and achievement are displayed on the screen every 5 seconds.
9. A beep goes off when a new coherence level is reached
10. 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
11. When the user feels satisfied with their session or wishes to end it, they press the selector to stop the session.
12. The user ends a session, and a summary view appears that include the following information: challenge level, percentage of time in different coherence levels (low, medium, and high), average coherence, length of session, achievement score, entire HRV graph
13. When the user ends their session the data from it will be stored in the history setting for future reference.
14. User presses the back button or menu button to continue using the device more or turns off the device completely.

Extensions
1a. If the battery charge is low, the device displays a warning on the session screen, prompting the user to recharge the device.

4a. The user can navigate to the settings tab to adjust the challenge level and breath pacer settings before a session, if needed.
4a1. The user can reset the device to its initial install condition, wiping all data and settings, through the menu options.
5a. If user presses selector button refer to UC3
5b. If anytime during a session
13a. The user can access the log/history tab in the menu to review past sessions and delete sessions if desired.
13a. If the user session is under 5 seconds, the data is not saved.

Use Case ID: UC2

Name: Battery Runout
Actor: User
Pre-condition: The HeartWave device is on and its battery power is low.
Post-condition: The device screen turns off.
Complete Scenario:
1. User uses the device.
2. Battery runs out.
3. device turns off.
Extensions:
3a) If the user was in the middle of a session.
3a.i) If the session time is more than 5 seconds.
The session info is saved and the screen turns off.
3a.ii) - If the session time is less than 5 seconds. Session data is lost and the device turns off.

Use Case ID: UC3

Name: Off button is pressed during a session
Actor: User
Pre-condition: HeartWave device is on
Post-condition: HeartWave turns off.
Complete Scenario
1. User uses the device and starts a session.
2. Before the session ends the battery is about to run out of power.
2a. If the session time is more than 5 seconds.
The session info is saved and the screen turns off.

2b. If the session time is less than 5 seconds. Session data is lost.

3- device screen turns off.

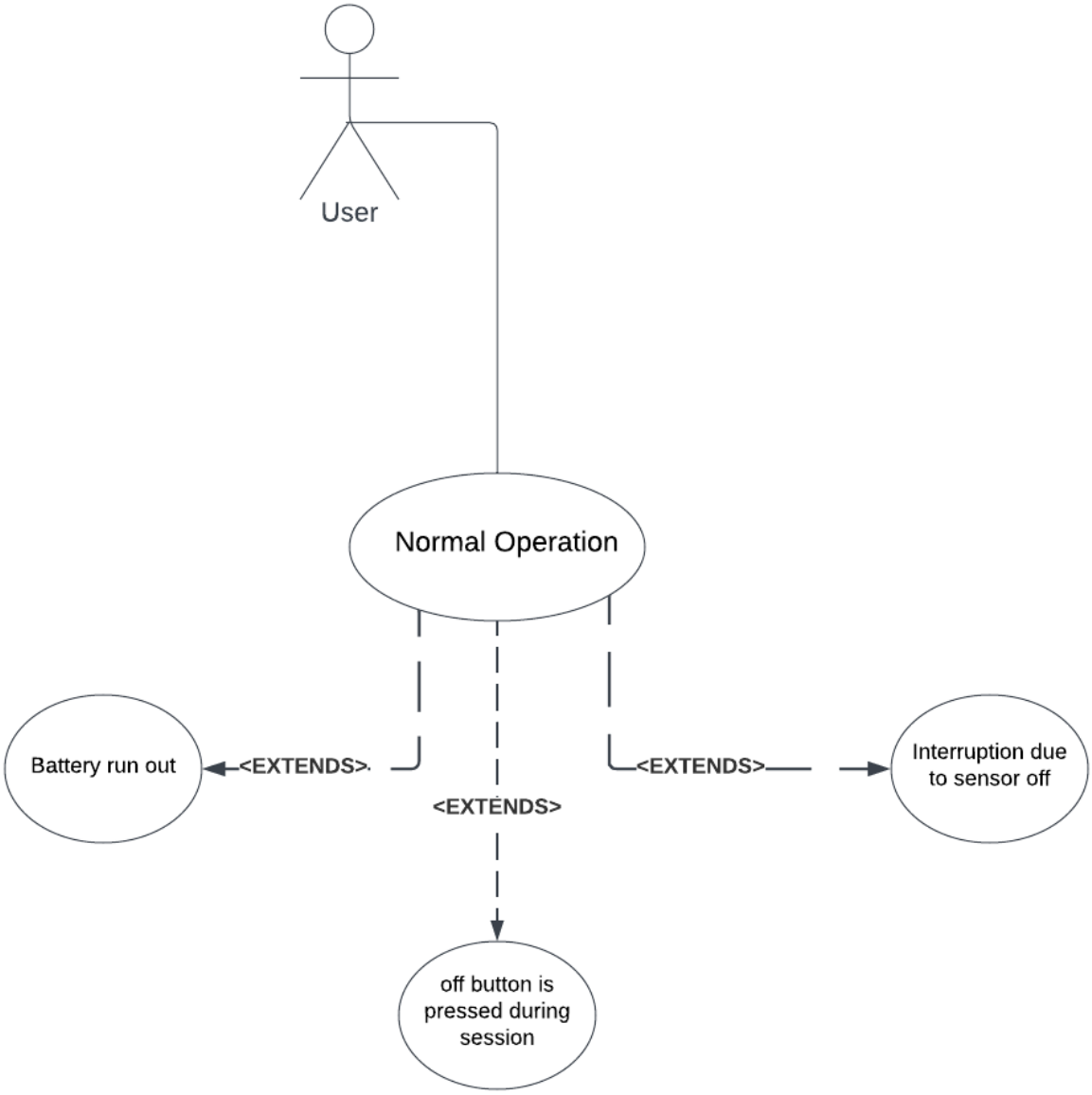
Use Case ID: UC4

Name: Interruption due to sensor off during session
Actor: User
Pre-condition: Sensor is connected to user
Post-condition: Session ends and a summary is displayed
Complete Scenario

1. Users starts the session
2. Sensor is disconnected or losses contact
3. Session is stopped.

3a. If the session was less than 5 seconds no data is saved and an empty summary window is displayed.
3b. If the session was longer than 5 seconds the session is saved and the data from the session is displayed on the summary screen.

Use case Diagram :



2.1 Textual explanation of our design decisions

The Use cases:

Use case ID: UC01

The main use case is the normal operation of the program, this begins once the power button is set to on, and then a number of functions need to run in order to display the graph, and coherence. Once the powerChange()

function is called, the changePowerStatus is also called, this is to turn on the display and call the necessary functions to life. Now with the screen turned on, and the device sensor is on and placed on the user's hand, we navigate through the subMenu() and once we click okay the session begins. A number of functions are also called now to display the session screen, the coherence graph, and give color to the led’s on the device. The breath pacer begins to move thanks to the BreathPacerMove() functions, and the graph begins to update, periodically for every 5 seconds. The generateData() function is called to actually plot and visualize the graph on the screen. The lightenCoherence lights are also called to enable the lights and to follow the coherence score of the application. Once the user clicks ok on the graph, the program halts to a stop and we call updateSummaryScreen() to show the session info and this is where we save the sessions data. We save the session as a current session, and plot the points while also adding this session to the device for memory. Once the user is satisfied with their results, they turn off the device in which the power is once again shut off and the device is no longer on.

Use case ID: UC02

The Battery runout use case starts when the battery level drops below 0 . The coherence lights are turned off by calling the darkenCoherncelLights() function . this function darks each color when the session is currently not in progress. Then return the user to the main menu and make a decision to save the current session or not by calling the navigateToMainMenu() function. In case the session current timer is larger or equal to 5 seconds , the system will save the session information by calling saveSessionData() function. If the current session timer is less than 5 seconds , the system will clean up the session and the information will be lost by calling cleanAfterSession(). Then , turn off the device by calling powerChange()

function , and charge the battery by calling setFullCharge() function , while changing the bar color to green by setting the QProgressBar stylesheet to green. Then set the value of the charger(QProgressBar) bar to the device battery level.

Use case ID: UC03: The interruption due to the sensor being off happens when a user is in a session, and the apply to skin function is called once again. The program then navigates back to the subMenu, and gives the user the summary screen, if the current session was less than 5 seconds then the device just discards that data and the cleanAfterSession function is called. However, if it is over 5 seconds, the saveSessionData() function is called, and all the information is once again saved by extracting the graph, adding the current session to the device and saving the points.

3.0 Video Link

https://www.youtube.com/watch?v=tgrpqQ-lylw

4.0 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 1,4	MainWindow.UI	Run the simulator and observe the heart symbol, use the admin combo box to change it on/off	When the apply to skin function is called, the heart light symbol is lit to indicate a active pulse reading

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 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.The menu options are displayed as default on the session screen. The menu could consist of the following options: start new session, settings, log/history.	Use Case 1, Use Case 3	MainWindow.UI Menu QCustomPlot	Run the simulator and observe the UI	Using QT's Built in interface, a user interface was created with a screen, off/on button, menu button, back button, four buttons and an ok button which acts as the selector. These buttons are used to navigate between the different menus and if a user wants to start a session the menu will also display a graph. The device will light up everything on the mainwindow once the power button is enabled, this depends on the powerchanged function.
3	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	Use Case 1	MainWindow.UI	Run the simulator, turn the sensor to true, power the device on, start a new session and observe the lights as you change the	The lights are lit once the function runSessionSim is called and the function lightenCoherenceLights is also called. The color depends on the coherence level being poor, medium and high.

				coherence level combobox.	
4	Press selector to initiate and end a session.	Use Case 1	MainWindow.UI	Run the simulator, press the power button, turn the sensor to true and press the OK button to let the session run and then press the OK button again to end it.	The OK button is connected to a navigateToSubMenu() function which observes if the state of the button changes to be clicked or not when the state indicates the button was clicked and if a user has selected the option to start a session/ a session is already running then it will initiate/end the session respectively.
5	Session screen must display the main HRV graph (HR vs time) with key metrics	Use Case 1	MainWindowUI, QCustomPlot	Run the simulator, turn the sensor to true, power device on, start a new session and observe the graph	This is done by using an external class, QCustomPlot to plot and construct a graph using the HR vs TIME metric. The graph updates every second a session is running and displays generated HRV data depending on the selected coherence level the graph shape will change.

6	The metrics on the screen include the current coherence score (numerical value), length (duration of session), achievement (total sum of coherence scores sampled every 5 seconds)	Use Case 1	MainWindowUI Session	Run the simulator, turn the sensor to true, power device on, start a new session and observe the total scores that are being sampled every 5 seconds	Every 5 seconds, we calculate the coherence score and update the label and text in the UI. The length of the session uses the Qtimer, and are displayed on labels.
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 seconds, adjustable in settings	Use Case 1	MainWindow.UI	Run the simulator, turn the sensor to true, power device on, start a new session and observe the breath pacer bar graph moving up and down	By utilizing the Progress bar widget we adjust the values of the progress bar to imitate the breathing in and out on a set interval between 1 second to 30 seconds. Default is set to 10 and is shown in the settings menu.
8	The breath pacer, 1-30 seconds, increases time interval between each breath, default at 10 seconds	Use Case 1	MainWindow.UI Device	Run the simulator, go to settings and adjust the breath pacer to anywhere between 1 and 30 using the left and right arrow keys of the device.	By setting the range of the breath_pacer in the progress bar widget, we can change the interval between each breath from 0,30 (setRange). These settings are reflected and are saved on the device.

9	When the user ends a session a summary view will appear that includes the following information: challenge level , percentage of time in different coherence levels (low, medium and high), average coherence, length of session, achievement score, entire HRV graph	Use Case 1	MainWindow.UI, Device, Session	Run the simulator, turn the sensor to true, power the device on, start a new session after 5 seconds, press okay and view the summary that pops up.	MainWindow class has a function that unhides additional info that is being calculated as long as the session is running such as the percentage of time in different coherence levels that are saved to the current session. This info is then populated on the UI labels.
10	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 1	MainWindow.UI, Device, Session, QCustomPlot	On the simulator, navigate through the settings and go to “History” to view the past sessions and show all data collected for them. Clicking on a specific session will also show the graph for it.	The sessions are stored as pointers, and once a user is finished a session it is and all of its data are stored in a vector inside the Device class. All plots’ points and info that was used to build the graph is saved to once again rebuild the graph after the session. Once history is selected they will be viewable with an option after each session to delete it.
11	An option to reset, wipe all data and restore the device to the initial install condition		MainWindow, Device, Session	On the simulator, go to settings, and reset the device to wipe all the data and restore the device to the	We set the current session to NULL, and invoke a function in our device class, “reset Settings”. This function sets the default time back to 10, the default

				initial install condition.	challenge to easy, and clears all sessions.
12	There is a battery charge indicator on the session screen	Use Case 1,2	MainWindow, Device	Start the power button, start a session and observe the battery slowly being drained.	There is a record of the power level of said device, and it slowly decreases. Once the battery level is dead, the screen turns black.
13	A beep goes off when a new coherence level is reached	Use Case 1	MainWindow	Running the program and observing the console for a “beep”	A Info is called once a new coherence level is reached. This is done in the same place the lights are being updated
14	The settings tab includes challenge level and breath pacer settings, there are 4 challenge levels for coherence, from beginner to advanced, for the user to choose	Use Case 1	Main Window	Running the program, going to settings, and enabling challenge level	The challenge value settings can be changed in the device class. By navigating through the settings, you can directly edit the challenge settings.