Textual Explanation for Design Decisions

We as a team decided to implement the observer design pattern which basically means we decided to use a bunch of connect (signals and slots) functions to simulate functionality of the device (heart rate variability monitor) by changing the behavior of the button when it is pressed. For our UI simulation we have a separate admin panel which will enable us to test our battery level (inspiration from DENAS), test data templates, allow us to simulate an interrupt session and lastly the entire hardware device with all buttons. Since most of the buttons/ labels and screens where being displayed and accessed in MainWindow we decided to use the MainWindow class as our main object/ entity class along with a separate menu class to help facilitate and create sub Menu (inspiration taken from DENAS) for challenge levels and breath intervals. The reason we decided to have them be implemented in a similar structure to menu format was primarily due to adding a dropdown to the menu screen with auto populated data for challenge level and breath interval.

For our graph data, the coherent one had an existing set data which limits the randomness of the graph and helps set them in a smoother flow. Unlike in the incoherent data we decided to use a RNG (Random Number Generator) since the incoherent data can be extremely irregular without showing any smoother flow of heart rate.

In the case of calculating our coherent score we have a random number generator ranging from 0 to 7 since 3+ is excellent depending on challenge level. When a new coherent score is reached we get a different/ similar color if the coherent score is in the specified range as per set by challenge levels and achievement score is just the sum of all coherent scores.

When a user decides to end a session, we decided to implement a bit of cellphone like functionality i.e.the user either hits on return to previous (back button) or menu button.

For the battery design decisions we decided to take a bit of inspiration from DENAS i.e. correlating at a constant rate of 0.05% every second. We also have the admin panel to help simulate low battery conditions when a user session is in progress or when the user is in the main menu panel.

For our history we decided to use a collection of data when the user ends a session, sensor is removed from skin or battery runs out it automatically stores it into the collection.