Major Design Decisions

Implementation:

We decided to use MainWindow as a controller/driver in this simulation in that PC and Device are not concrete objects but rather portrayed by numerous QObjects in MainWindow.ui/MainWindow. This is to streamline the interaction between different elements for less obfuscation and possible points of error. In other words, by using MainWindow to portray both PC and Device via separate Stacked Frames/ Widgets and QObject groups on the same MainWindow the main form of class information exchange is between MainWindow and Session. For the purpose of the simulations we saw this to be perfectly fine as we felt adding more classes would simply overcomplicate things. Resources such as QTimers are also easier to handle in this design and given that QTimers and signals and slots are the main driving force in our design this design simply made the most sense.

Our main design patterns used were states and qt signals and slots. While the device technically had state elements such as power on and off a battery level that were in the form of numerous attributes in MainWindow, we didn't feel it played as crucial a role as the Session States did and as such thought only Session deserved a proper state diagram. SessionStates were the main driving force in our sessions events and loops, alongside QTimer signals and slots. Before further explanation on the state, an explanation of the slots and signals are necessary. Session has 2 timers, progressTimer on 1 sec interval, and electrodeTimer on 1/16s interval. Session also has a slot that progressTimer connects to called baselineCalc, and a slot that electrodeTimer connects to called treatElectrode. On a purely a session level view, both of these slots have logic that simulates pseudo counter controlled loops where the counters are required aspects of a certain session round like 16 total pulses, or a segment of time for baselineCalcs, or rounds completed, etc. Signals and Slots are extraordinarily helpful with this because it allows us to interrupt at any signal iteration and resume, something that would be difficult if it were a normal for loop. For instance, we can start and stop at any point in either portion of the treatment this way. This is how both these slots work. The States are used to help decide which session event should currently be active since progressTimer will never be stopped as it is also connected to the updateCountdown which tracks the timer going down to 00:00. As such, porgressTimers signal still reaches the slot for baselineCalc, but it will only begin baseline calc if the state is ANALYZING. Session also implements a history state, allowing us to know where the session is paused so it can resume from where it left off instead of restarting any cycles.

IMPORTANT NOTES ON SEQUENCE DIAGRAMS:
For instance of terminateSession(bool), the boolean communicates whether the device must turn off or not after that function call. For example in the battery sequence we want to return to the main menu. So it's false. But for connection loss given its longer than 5 min, we want to turn off the device so the bool is true.

Also please note that in the main treatment sequence the updateCountdown() loop is everlasting concurrently until session is over. It runs in parallel as you travel down the sequence diagram.

One last note, please recall from the beginning of this document that the PC is a part of the MainWindow alongside the device, which is why in The therapy viewing on PC sequency we are calling functions on the mainwindow for both the PC and Device. All PC button functions have PC in the name.