Design decisions:

1) Justification for Keeping Alerts and Settings separate from options and on contentwidget instead which serves as the main window Ui, is to increase efficiency making it faster to access as alerts and settings are frequently used throughout the process whether you want to view data or alerts on the insulin pump. Having both inside options would just make the process take more time and be less user friendly. And this design decision was approved by TA Karim and Professor Radonjic stating “Prof said your UI looks good, no need to change it”.
2) Class diagram display of signals and slots: After looking through the UML course resources, we could not find any mentions of how to document signals and slots so we decided to add them as a sub-section of the function part of its corresponding class.
3)
   a) Model: statusmodel class serves as the data model. Manages states like battery level/ charging status/insulin on board.
   b) View: Ui files like statusbar.ui that defines the visual layout. View shows the data from the model.
   c) Controller: MainWindow acts as the main controller, handling routing between different Ui interfaces. It responds based on what the user inputs and changes the model that is eventually shown in the view.
4) Singleton Pattern: Multiple classes implement the Singleton pattern:StatusModel::getInstance(), StatusBar::getInstance(), AuthManager::getInstance()
5) Observer pattern: statusmodel emits signals like batteryLevelChanged. Observers such as the statusbar connect to these signals using slots like onBatteryLevelChanged.
6) Facade Pattern: We have a single instance of battery which is abstracted by the statusmodel facade, that way we can maintain a single instance of battery and only interact with only essential features that it provides so that we don’t directly change the drain timer for example. statusmodel includes only the necessary methods required by other areas of the code and allows us to alter battery’s implementation in general without affecting how other areas in the code use it.