ID	Requirements	Related Use-case	Fulfilled by	Test	Description
1	Ensure the application is intuitive and user-friendly	N/A	MainWindow.ui	Run the simulator in Qt to observe the ui.	
2	The application interface contains buttons and display	N/A	MainWindow.ui	Run the simulator in Qt to observe the ui.	Using QTs built in user interface framework, the physical RaDoTech system was replicated. Also, all buttons are clickable and functional.
3	Ensure the system can handle a growing number of users and data points	Creating a new RaDoTech Profile	MainWindow	Run the simulator in Qt to observe the ui.	The RaDoTech code was implemented using easily modifiable arrays to track/store persistent user and data memory while running simulator The MainWindow class initializes and creates all necessary objects,
4	Create, update, and delete user profiles. Each device should support multiple profiles (up to five)	Creating a new RaDoTech Profile	MainWindow, User	Run the simulator in Qt and create maximum number of test user profile	The User class stores all the collected user information. The User object is initialized by MainWindow which controls user profile creation and deletion
5	Interface with RaDoTech hardware to collect health data.	RaDoTech Health Monitoring Device	AppWidget	Select a user and from the sidebar menu select Measure page.	Using QTs built in user interface framework, the physical RaDoTech system was replicated and is implemented using a stacked Widget QObject alongside a sidebar panel for user navigation to stacked widgets.

6	Show the device contacting or not contacting the skin.	RaDoTech Health Monitoring Device	AppWidget	Start a scan and select scan hands or feet button, then observe the ui.	The AppWidget class contains the scanning implementation of the RaDoTech device. It checks for device contact with skin before every scan, if no skin contact sensed, it aborts scan, updates ui then retries the scan
7	Process raw data to generate health metrics as you understand it	RaDoTech Health Monitoring Device	AppWidget , Data	Select a user and from the sidebar menu select Data page and toggle data.	The Data class replicates live raw reading from a sensor using a random number generator, generating values from an acceptable range. It passes the raw data into a function, to be processed to a user friendly, visually appealing format, with a color legend to aid user.
8	Display health metrics in an easy-to-understand, visual format within the application	RaDoTech Health Monitoring Device	AppWidget, Data, BarChart	Select a user then complete a scan. From the sidebar menu select Data page and toggle graph	The visual format is implemented in the BarChart Class. The barChart QWidget is initialized by the AppWidget class and receives data points to be displayed in bar chart
9	Provide a place-holder for specialists' recommendation	N/A	AppWidget	Select a user and from the sidebar menu select Data page, then toggle to recommendation.	This is implemented by the AppWidget class as a QStackedWidget.
10	Store and allow users to access historical health data	View Scan History	AppWidget, User, Data	Select a user and from the sidebar menu select history page.	The user historical data is implemented and stored by the User class in a QList<Data> QWidget

11	Show charge depletion and low power indication.	Charge RaDoTech Device	AppWidget	Start a treatment and observe the battery icon: battery level will lower with each completed scan. Select charge from sidebar menu to recharge	The battery replicated in the UI using a Qwidget and implemented by AppWidget class It keeps consistent record of the power level of the device, to track low battery. It contains QTimer it updates the UI display. As each scan is run, the battery level will consistently decrease until charged