

# COMP 3004 - Final Project

Group Members:

- Matthew de Sousa
- Rachel Wong
- Evan Raz
- Jaedong Koh

## Use Cases:

### Use Case: Create Profile

Primary Actor: User

Scope:

Level:

Stakeholders and Interests:

- User: Wants to create a profile.

Preconditions:

- The system is powered on.
- The system is displaying the login window.

Success guarantee:

- A profile is created and saved in the system.

Main Success Scenario:

1. The user selects "Create Profile."
2. The system prompts the user to enter the required information.
3. The user inputs the required information into the designated fields.
4. The user selects "Create Profile" to finalize the profile creation.
5. The system validates the input.
6. If validation passes, the system creates the user profile.
7. The system returns to the login page.

Extensions:

4. a. The system's profiles list is full
  1. The system displays a message indicating the maximum number of profiles has been reached.
  2. The system returns to the login page.
6. a. The system fails to validate the input:
  1. The system highlights the relevant input fields indicating the issue.

2. The user corrects the error.
3. Resume at step 4.

### **Use Case: Update Profile**

Primary Actor: User

Scope:

Level:

Stakeholders and interests:

- User: Wants to update their information.

Preconditions:

- The system is powered on.
- The user already has a profile created.
- The system is displaying the main window.

Success guarantee:

- The profile's information is updated and saved in the system.

Main Success Scenario:

1. The user selects "Profile."
2. The system displays the current user profile information.
3. The user makes the desired changes to the profile information.
4. The user selects "Confirm Changes."
5. The system validates the input.
6. If validation passes, the system updates the user profile.

Extensions:

5. a. The system fails to validate the input:
  1. The system highlights the relevant input fields indicating the issue.
  2. The user corrects the error.
  3. Resume at step 4.

## **Use Case: Delete Profile**

Primary Actor: User

Scope:

Level:

Stakeholders and interests:

- User: Wants to delete a profile.

Preconditions:

- The system is powered on.
- The user already has a profile created.
- The system is displaying the main window.

Success guarantee:

- The profile is permanently deleted from the system.
- All associated data is removed from the system.

Main Success Scenario:

1. The user selects "Profile."
2. The user selects "Delete Profile".
3. The system prompts the user with a warning message and to input their password.
4. The user presses the "Delete My Profile" button.
5. The system validates the input.
6. If validation passes, the system removes the profile and all associated data.
7. The system returns to the login screen.

Extensions:

5. a. The system fails to validate the input:
  1. The system cancels the deletion.
  2. The system displays a cancellation message.
  3. The system closes the deletion prompt.

## **Use Case: Collecting Data**

Primary Actor(s): User

Stakeholders and interests:

- User: wants to collect data from 24 measurement points.

Preconditions:

- User profile exists and is selected for the session.
- The device is powered on and ready for the scanning.

Success guarantee :

- Data for all 24 points are collected and stored in the system.
- Data is ready for processing

Main success scenario:

1. User starts a new health scanning session.
2. System prompts the user to position the device at the measurement point.
3. System verifies skin contact at the specified point.
4. System collects the data for the measurement point.
5. User lifts the device off the skin.
6. User positions the device at the next measurement point.
7. Repeat step 2-6 until all 24 measurement points are scanned.
8. User can enter any additional information.
9. System stores the collected data in the user's profile.

Extensions:

3. a. User does not lift the device off the skin :
  1. System waits for the user to lift the device off the skin and provides a notification.
4. a. Skin contact is not detected:
  1. System waits until proper skin contact is established and provides a notification.

## **Use Case: Processing Data**

Primary Actor(s): System

Stakeholders and interests:

- User: wants accurate health metrics derived from collected data.

Preconditions:

- Data for all 24 measurement points has been successfully collected and stored.

Success guarantee :

- Raw data is accurately processed into health metrics
- Processed data is securely stored and ready for visualization.

Main success scenario:

1. System retrieves all 24 measurement points data.
2. System compares the value to the predefined thresholds for normal, below norm and above norm ranges.
3. System classifies each data point into one of these categories.
4. System stores the processed data and classifications in the user's profile.

## **Use Case: Visualizing Reading Data**

Primary Actor(s): User

Stakeholders and interests:

- User: wants to see their health from understandable visual representation

Preconditions:

- Data must be collected and stored.

Success guarantee :

- The user is able to view historical readings through a visually-pleasing chart.

Main success scenario:

1. User clicks on the Charts tab.
2. The system generates and displays a detailed graph reflecting the selected area of interest.

Extensions:

2. a. There is not enough data to fill the chart:
  1. Visually distinct empty bars are added for padding, with a “No Data” tooltip
2. b. There is too much data to fill the chart:
  1. Surplus data points are not shown, and the chart maintains a 20-session rolling data window.

### **Use Case: Displaying and Updating Past Notes**

Primary Actor(s): User

Stakeholders and interests:

- User: wants to view and edit their notes from past sessions

Preconditions:

- None

Success guarantee:

- The user is able to view and modify any note they have previously saved.

Main success scenario:

1. User clicks on the Notes tab.
2. The system displays the latest note, with editable fields.
3. The user uses the arrow buttons to browse through to a note of interest
4. Should they wish, user edits any or all of the notes fields
5. Should they wish, the user presses the save button to update the note

### **Use Case: Battery use**

Primary Actor(s): User

Stakeholders and interests:

- User: wants to use the device both on battery or while plugged in

Preconditions:

- Battery has initial charge

Success guarantee:

- The device will charge when plugged in but remain operational when unplugged
- Should the battery die, the device will shut down gracefully

Main success scenario:

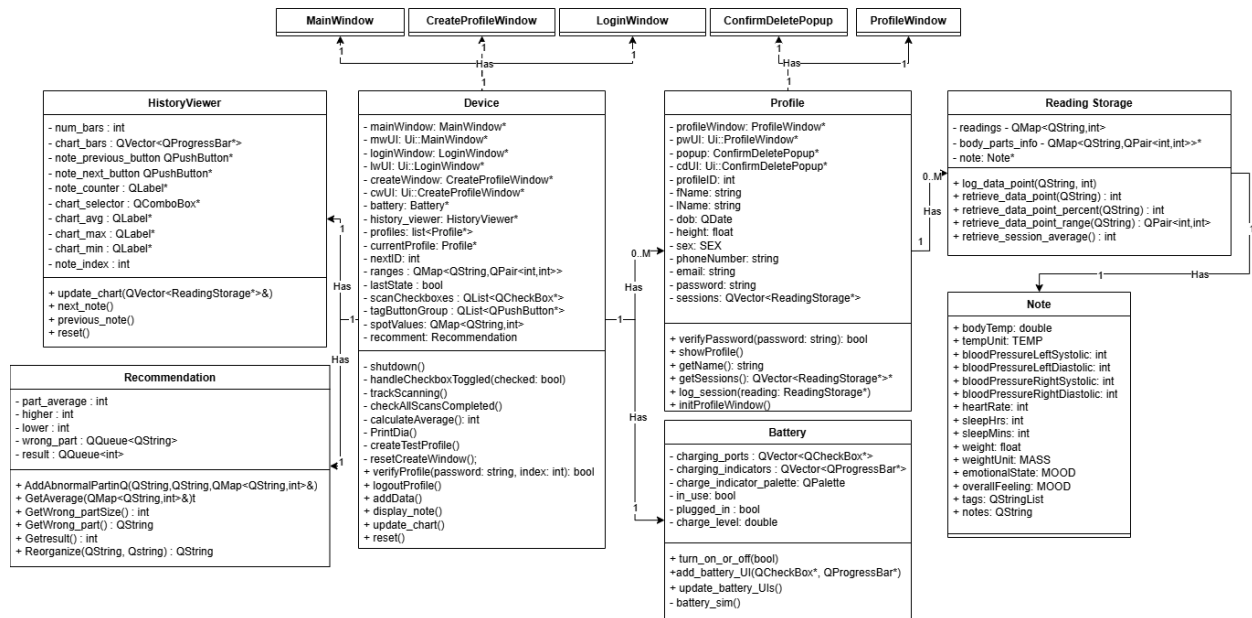
1. User plugs in the device, and it charges
2. User unplugs the device, and device remains functioning under battery power

Extensions:

2. a. The battery charge is low
  1. The UI indicated a low battery
2. b. The battery depletes:
  1. The device notifies the user and performs a graceful shutdown

# Design documentation:

## UML Class Diagram



### Device Class

- The device class acts as a container for all other classes, creating a centralized system flow controller.
- The device class contains all connections of signals and slots between any of its members.

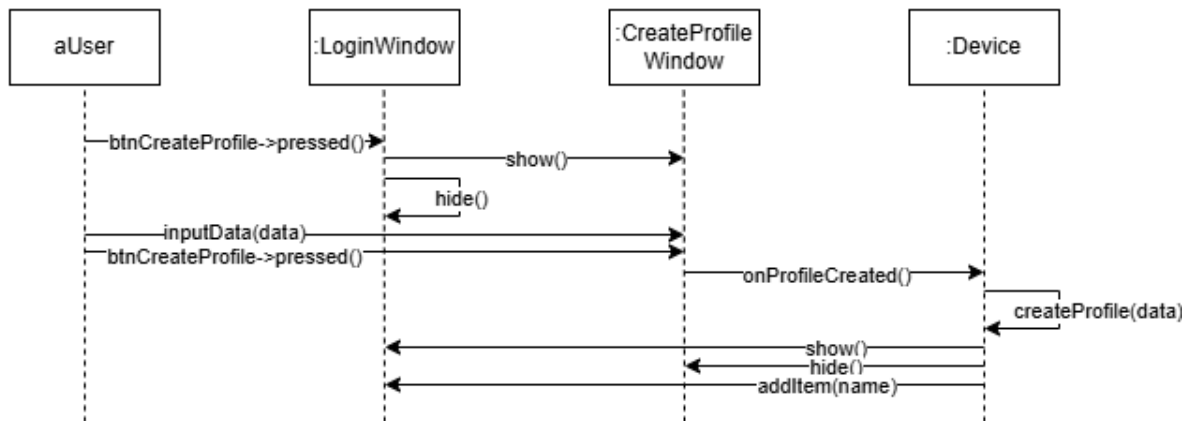
### Profile Class

- The profile class acts as a container for the user's information and data storage, and it contains its own profile window.

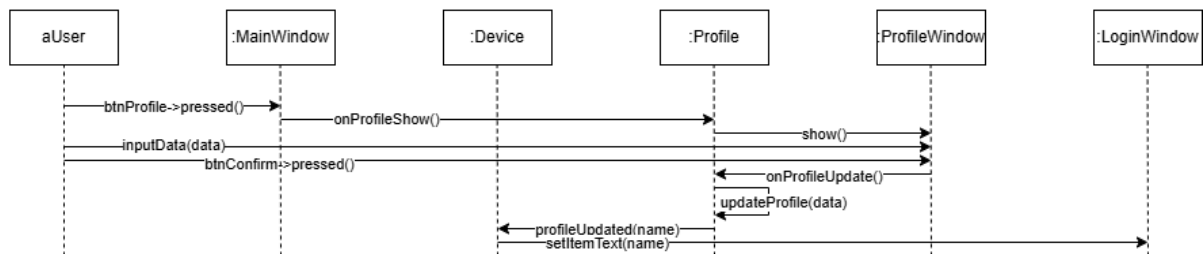


# UML Sequence Diagrams

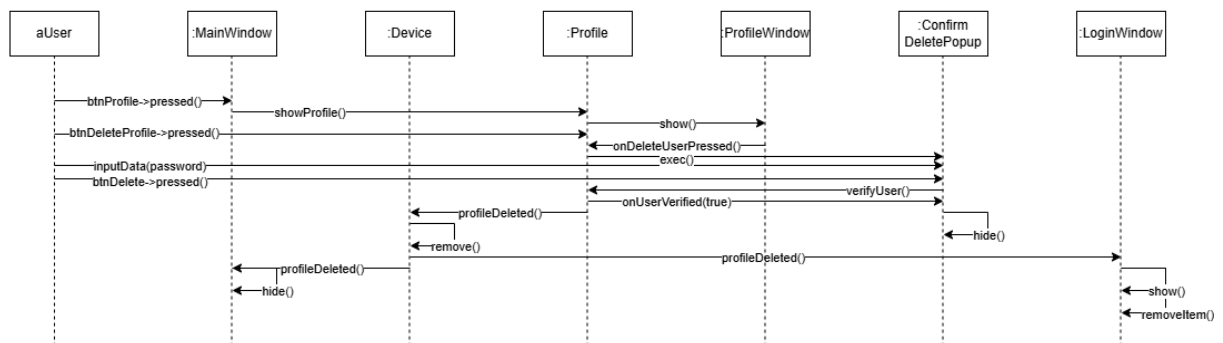
## Creating a Profile



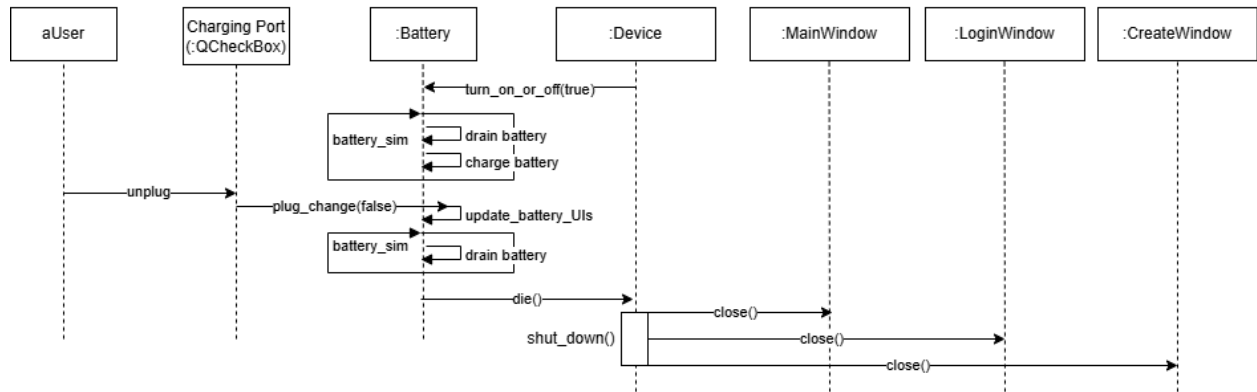
## Updating a Profile



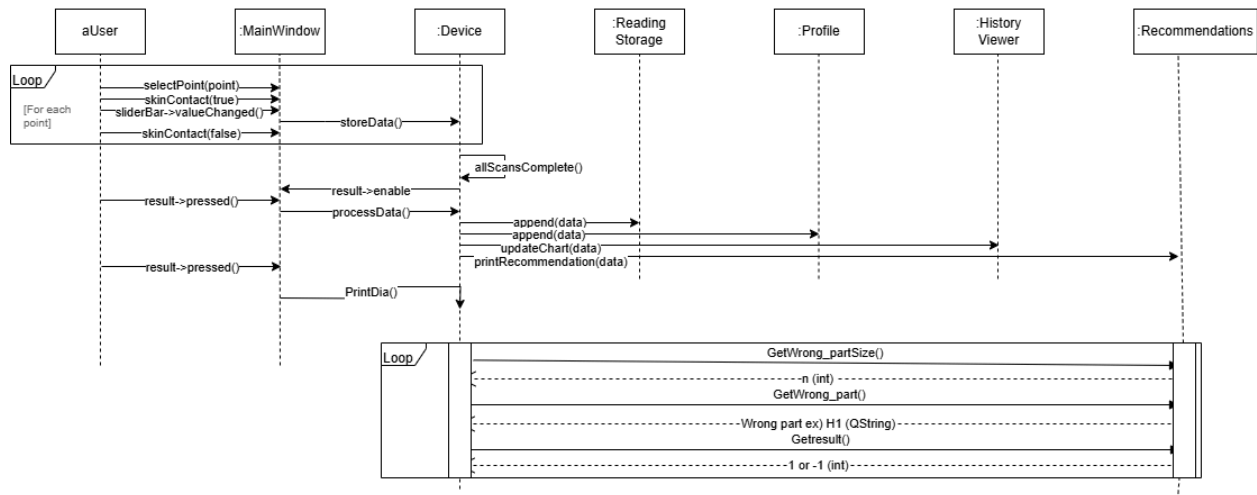
## Deleting a Profile



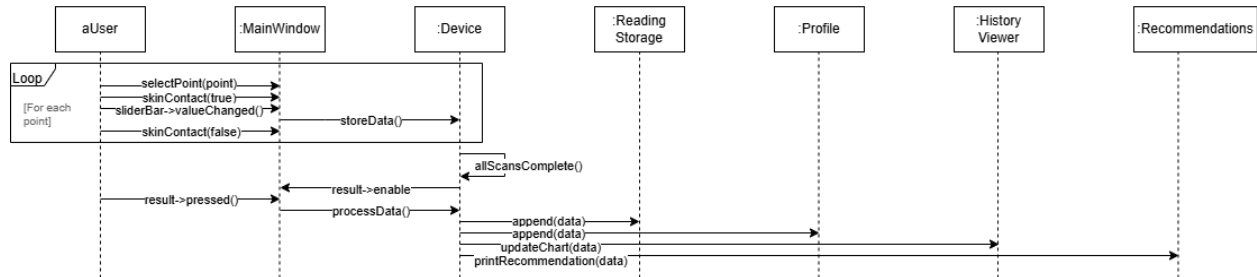
## Battery



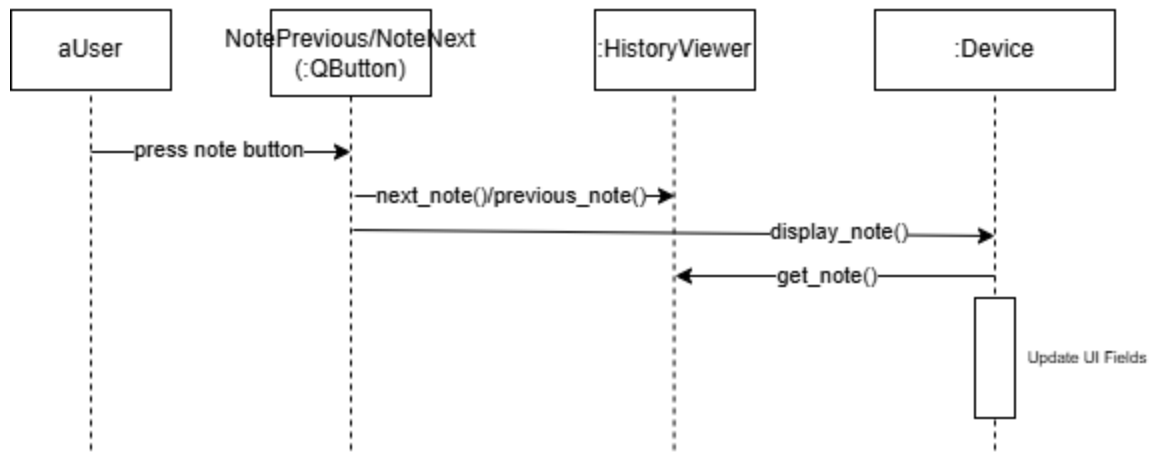
## Print Recommendation



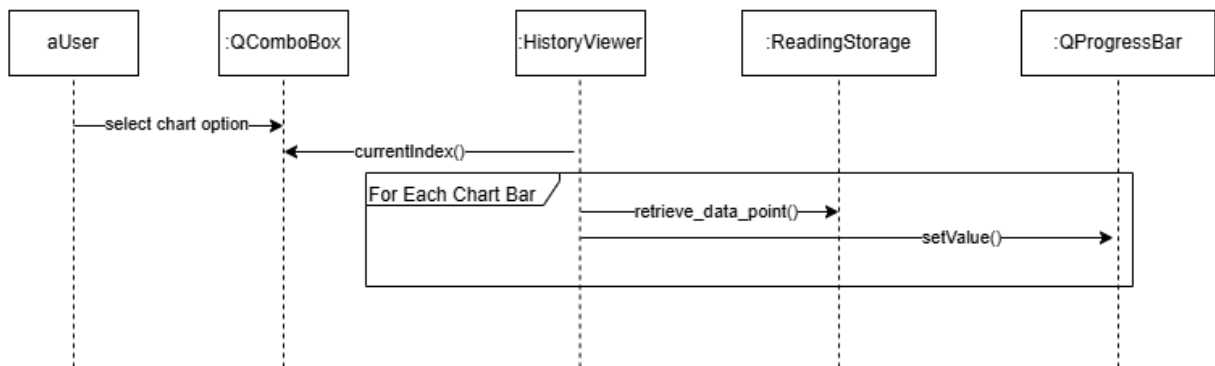
## Data Collection



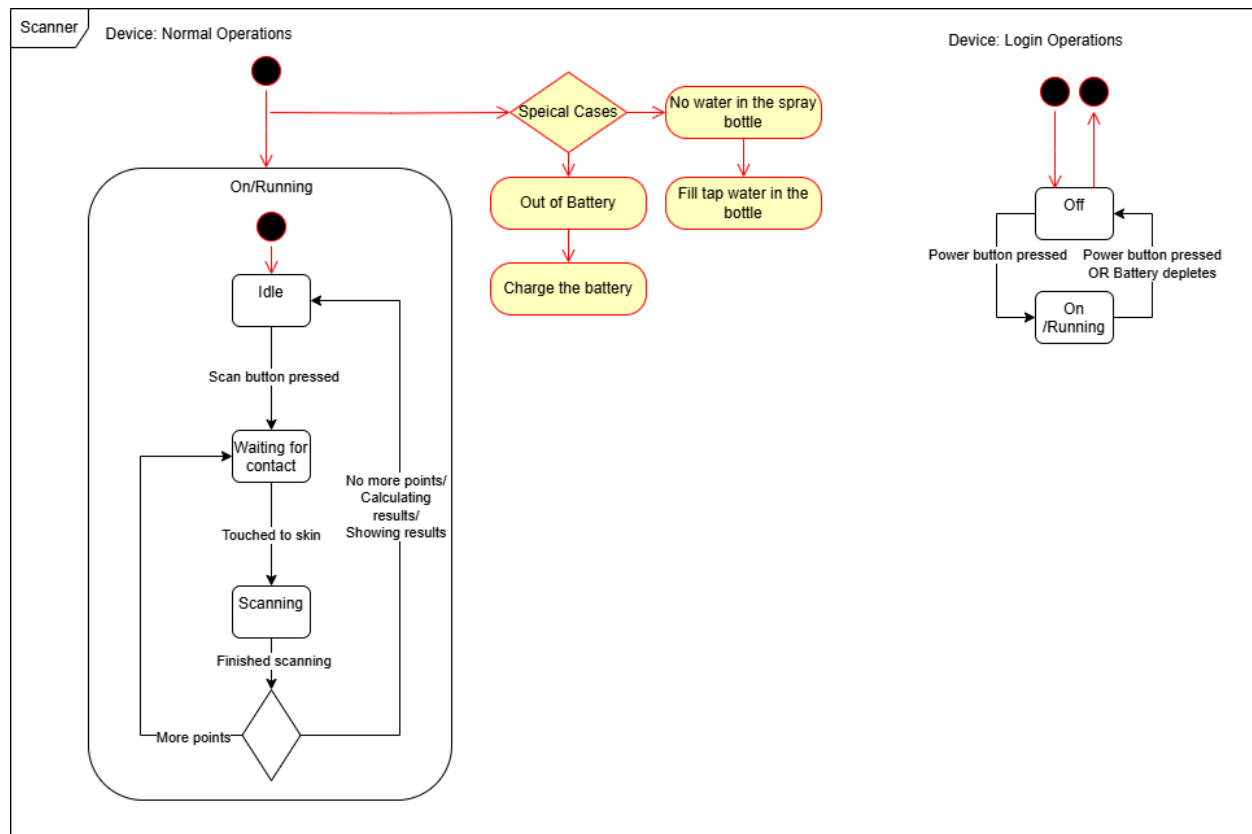
## View Notes



## Visualize Reading Data



# State Machine Diagram



## Design Choices Explained

### Design Patterns:

- **Mediator and Observer**
  - The device class acts as a mediator and observer class. Mediating between classes and windows and forwarding communications from one class to multiple others and observing the signals sent by classes and handling them appropriately.
  - The profile class also acts as a mediator between its members. The currently logged-in profile mediates all communication to itself from the device and its members.
- Singleton - Allowing only one profile to be logged in simultaneously avoids conflicting data with multiple profiles.
- Facade - The history viewer class acts as a facade for the data, masking the complexity of the data and providing a simplified interface in the chart.

### Data Processing Explanation:

After conducting research, we realized there are two approaches for generating health metrics from raw data collected:

1. Fixed Threshold Method:
  - a. If the measured value is below  $40\mu A$ , it shows the measurement is below norm.
  - b. If the measured value is above  $80\mu A$ , it shows the measurement is above norm.
  - c. Otherwise, if it is within the range of  $40\mu A$  to  $80\mu A$ , it shows the measurement is normal.
2. Dynamic Range Method:
  - a. The average Ryodoraku is calculated by summing all 24 measurement and dividing by 24.
  - b. If the measured valued is  $average + 20\%$ , the measurement is above norm.
  - c. If the measured valued is  $average - 20\%$ , the measurement is below norm
  - d. Otherwise, the measurement is normal.

Source: <https://pmc.ncbi.nlm.nih.gov/articles/PMC6831389/#:~:text=First%2C%20less%20than%2040%20%CE%BCA,an%20individual's%20average%20Ryodoraku%20current.>

Combining our research with the Ryodoraku Chart from the specification, we determined that the second approach aligns with RaDoTech's data processing algorithm. Therefore, we chose to use the dynamic range method.

**Graceful Shutdown Explanation:**

A main goal of graceful shutdown is that no data gets partially or incompletely saved. To ensure this, all data gets saved in batches rather than as it is read. For example, all electrical readings are saved into a temporary array. Only when all readings are completed are they saved to the user's profile all at once. This prevents power failures from causing partial logs, which would lead to data corruption and disrupt any future trend analysis or history viewing.

## Requirements Traceability Matrix

ID	Requirement	Related Use Case	Fulfilled by	Test
1	Users must be able to create a profile by entering personal information.	Create Profile	CreateProfileWindow.ui Device	Enter the profile information and verify the profile is created successfully.
2	Users must be able to update their existing profile information.	Update Profile	ProfileWindow.ui Profile	Open the ProfileWindow and change each line of information, verifying the information is updated through the application.
3	Users must be able to delete their profiles, permanently removing all associated data.	Delete Profile	ProfileWindow.ui ConfirmDeletePopup.ui Profile Device	Open the ProfileWindow, begin to delete the profile and verify the profile and all data were deleted successfully.
4	The system must collect data from 24 different measurement points during each session.	Collect Data	MainWindow.ui Device ReadingStorage	Manually test all 24 points for successful data collection. Print all the data in the terminal for viewing, or view individual point values in the charts tab later
5	Users must be able to enter additional information after the data collection phase.	Displaying and Updating Past Notes, Collect Data	MainWindow.ui Device ReadingStorage Note	Complete a data collection session, navigate to the notes tab, and manually test each field, verifying the note is saved.
6	Collected data must be stored in the user's profile.	Collect Data	Profile, ReadingStorage, Note	Complete a data collection session, log in and out of the profile, verifying each point of data is successfully stored and dependent on the currentProfile.

7	The system must process raw data into health metrics.	Processing Data	Device Recommendation ReadingStorage	Complete a data collection session, navigate to the charts and recommendations tabs to verify each point of data is processed successfully.
8	The system must generate health insights from processed data.	Processing Data, Visualizing Reading Data	MainWindow.ui Device Recommendation	Complete a data collection session, navigate to the charts and recommendations tabs to verify each point of data is processed successfully.
9	Users must be able to view historical readings through visually appealing charts or graphs.	Visualizing Reading Data, Processing Data, Collecting Data	MainWindow.ui Device HistoryViewer	Complete a data collection session, navigate to the charts and notes tabs to verify each point of data is processed successfully.
10	The device must indicate battery status to the user.	Battery use	MainWindow.ui Battery	Test the program as normal and observe the battery status indicator decreasing.
11	The device must be able to charge the battery.	Battery use	MainWindow.ui Battery	Test the program as normal and observe the battery status indicator decreasing. Once the battery is close to empty, check the charging indicator and observe the battery status indicator increase.
12	The device must shut down gracefully upon battery depletion	Battery Use	Battery, Device	Allow the battery to deplete and observe the program exit