[Team 8] COMP 3004 - Final Project Documentation

RaDoTech Use Cases:

Creating a User Use Case (UC1):

Primary Actor:
- User

Precondition:
- User has the RaDoTech application installed and opened on their device

Main Success scenario:
1. From the home page the user selects create profile
2. User inputs their information along with an email and password
3. User is created

Postcondition:
- Device enters the dashboard of the created user and is waiting for an action from the user

Creating a Profile Use Case (UC2):

Primary Actor:
- User

Precondition:
- User has the created a user for the RaDoTech application and is in the dashboard

Main Success scenario:
1. From the user dashboard page the user opens the side menu and goes to the profile tab
2. User selects ‘new profile’, adds information then selects the add profile button
3. Profile is created

Postcondition:
- Device stays on the profile page after the profile is created and is waiting for an action from the user

Extensions:

2a. Maximum profile count reached.
- The user must remove an existing profile on their device to be able to create a new profile

Updating a Profile Use Case (UC3):

Primary Actor:
- User

Precondition:
- User has an existing profile in their RaDoTech application

Main Success scenario:
1. From the dashboard the user opens the side menu to select the profiles page
2. User has selected the desired profile and selects the option to update information
3. User updates their profile information
4. Profile is updated

Postcondition:
- Device stays on the profile page after updating the profile and is waiting for an action from the user

Deleting a Profile Use Case (UC4):

Primary Actor:

- User

Precondition:

- User has an existing profile in their RaDoTech application

Main Success scenario:

1. From the dashboard the user opens the side menu to select the profiles page
2. User has selected the desired profile and selects the option to delete the profile
3. Profile is deleted

Postcondition:

- Device stays on the profile page after deleting the profile and is waiting for an action from the user

Scanning Process Use Case (UC5):

Primary Actor:

- User
- RaDoTech Scanning Device

Precondition:

- The application is on the dashboard of the logged in user

Main Success Scenario:

1. User selects the scanning option from the dashboard
2. The user selects the profile they wish to scan with
3. User places the scanning device on the skin to measure the conductance
4. RaDoTech scanner senses skin and begins to output an electrical current.
5. RaDoTech scanner informs the device of conductance reading and advances to the next meridian point.
6. Step 4 and 5 repeat until all the meridian points are scanned, then press finish scan.
7. The ScanSession is completed

Postcondition:

- The scan session is saved to history and the device redirects to the diagram/results page

Extensions:

4a. Use case can be stopped at any time if the user presses the back button which will redirect the user back to the dashboard of the user.

View Scanned History Use Case (UC6):

Primary Actor:

- User

Precondition:

- The application is on the dashboard of the logged in user

Main Success Scenario:

1. User opens the side menu
2. User selects the History tab
3. The device displays all saved scans of the selected profile. If no scans were saved, nothing is displayed

Postcondition:

- Every scan from all profiles is displayed to the user.

Display Diagrams for Data Visualization Use Case (UC7):

Primary Actor:

- User

Precondition:

- A scan session was completed

Main Success Scenario:

Primary:

1. From the dashboard the user selects the detail button pulling data from the recent scan
2. Application displays the body diagram with organs.
3. User selects the chart tab.
4. Application displays the health metric in bar chart form.

Secondary:

1. From the dashboard user opens the side menu
2. User selects the History tab
3. The device displays all scans from all profiles
4. Selecting any of the previous scans, the device will redirect to the diagram pages with the results from that past scan.

Postcondition:

- The diagrams are displayed to the user for visualization.

Display Indicators Use Case (UC8):

Primary Actor:

- User

Precondition:

- The application is on the diagrams page after a scan session is finished.

Main Success Scenario:

1. User selects the indicators tab
2. The device displays all the indicators relating to the functionality of the body

Postcondition:

- All health indicators for the selected profile are displayed to the user.

Display Recommendations Use Case (UC9):

Primary Actor:

- User

Precondition:

- The application is on the diagrams page after a scan session is finished.

Main Success Scenario:

1. User selects the recommendation tab
2. The device displays all the recommendations

Postcondition:

- All recommendations for the selected profile are displayed to the user.

Low Battery Use Case (UC10):

Primary Actor:

- User
- RaDoTech Scanning Device

Precondition:

- The device battery must have charge and the application is on the dashboard of the desired profile

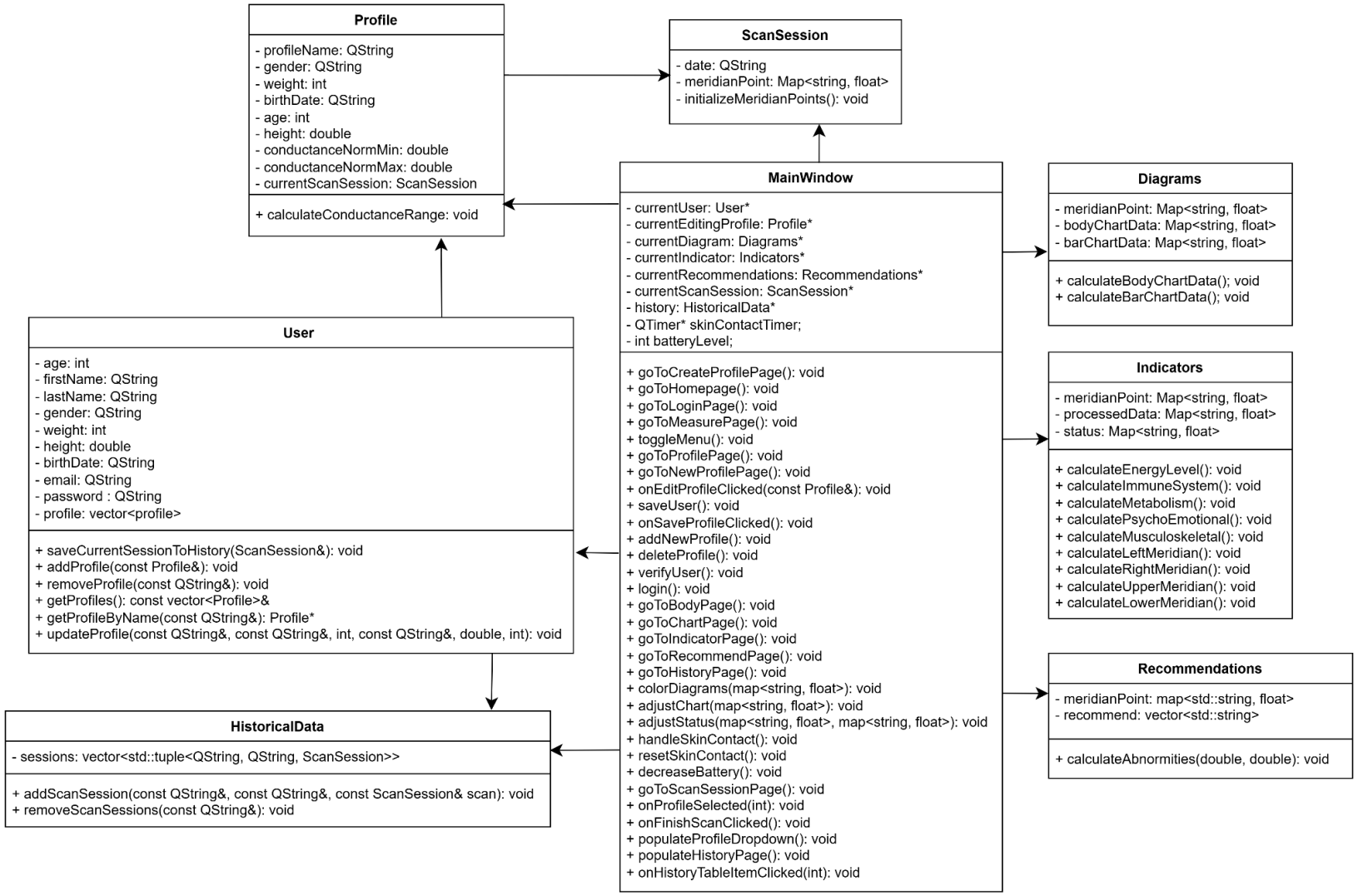
Main Success Scenario:

1. User selects the scanning option from the dashboard
2. The user selects the profile they wish to scan with.
3. User places the scanning device on the skin to measure the conductance
4. RaDoTech scanner senses skin and begins to output an electrical current.
5. RaDoTech scanner informs the device of conductance reading and advances to the next meridian point.
6. The battery of the device incrementally decreases, as steps 4 and 5 are repeated.
7. Step 4 and 5 repeat until all the meridian points are scanned or battery finishes.
8. The device stops due to no battery.

Postcondition:

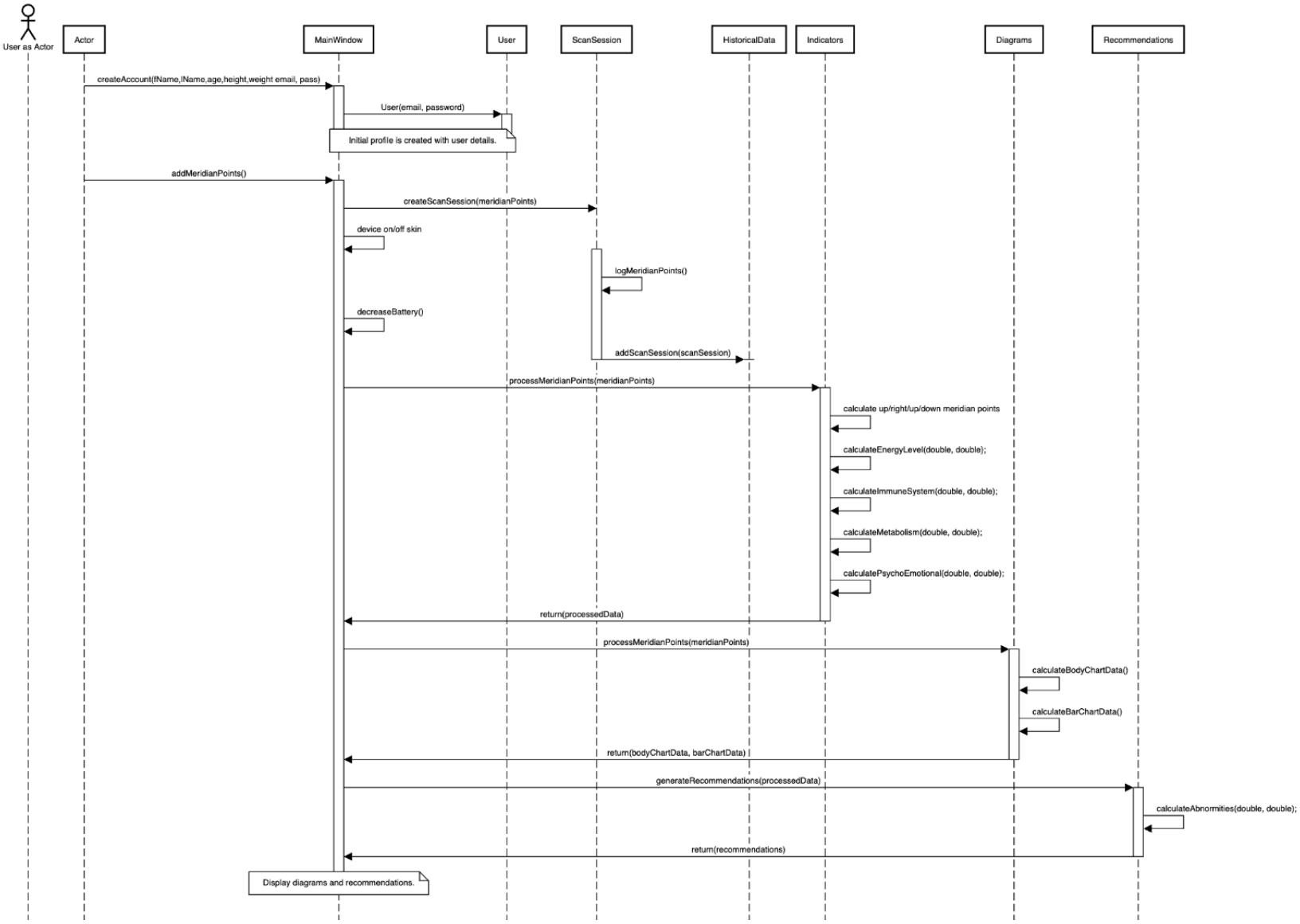
- The application shuts down after the battery is finished.

UML Class Diagram

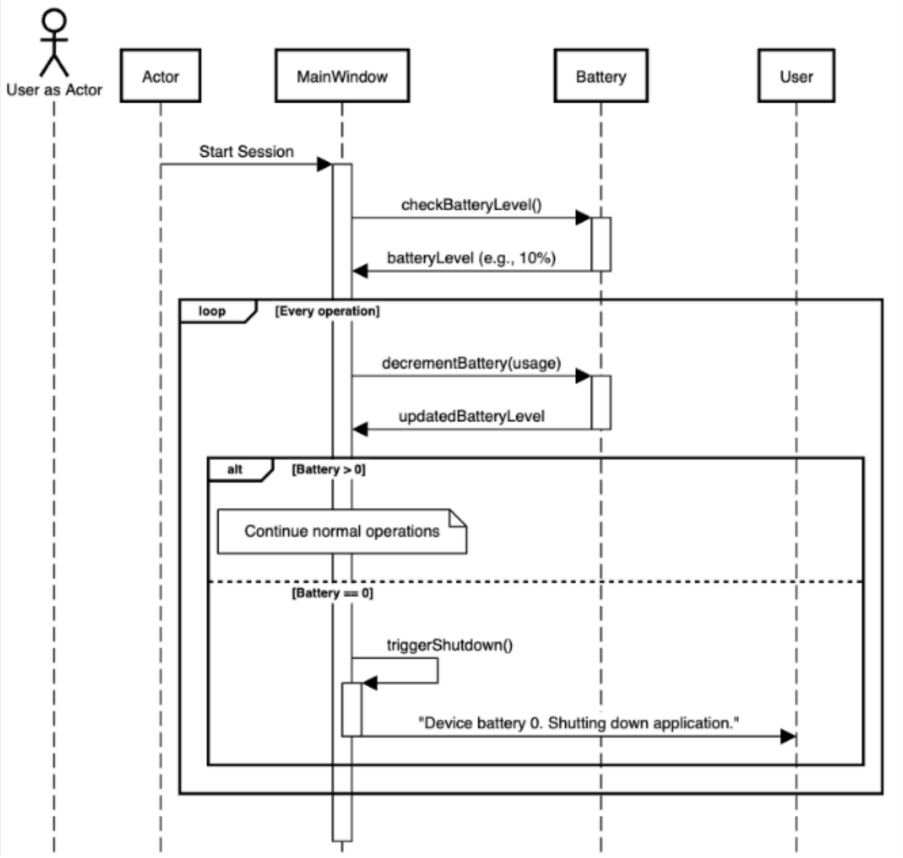


Sequence diagrams

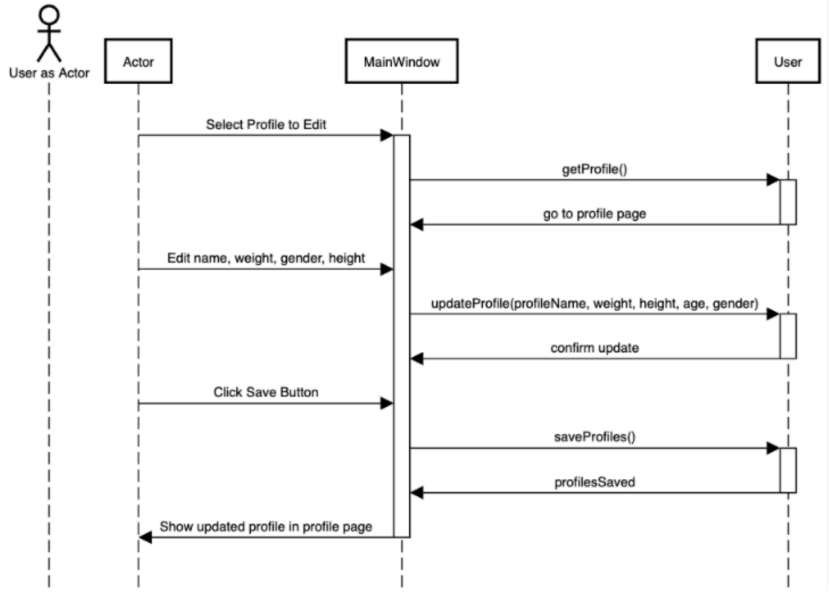
End to end



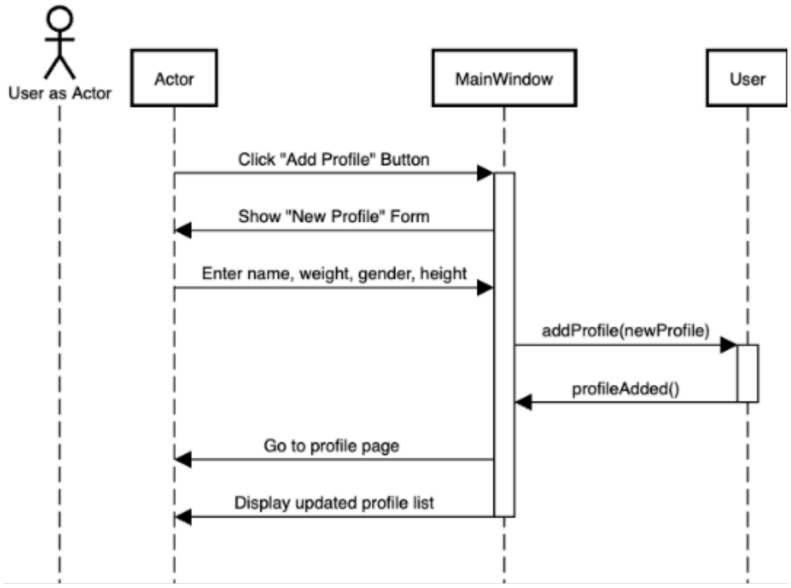
Battery draining to 0 and application shutdown



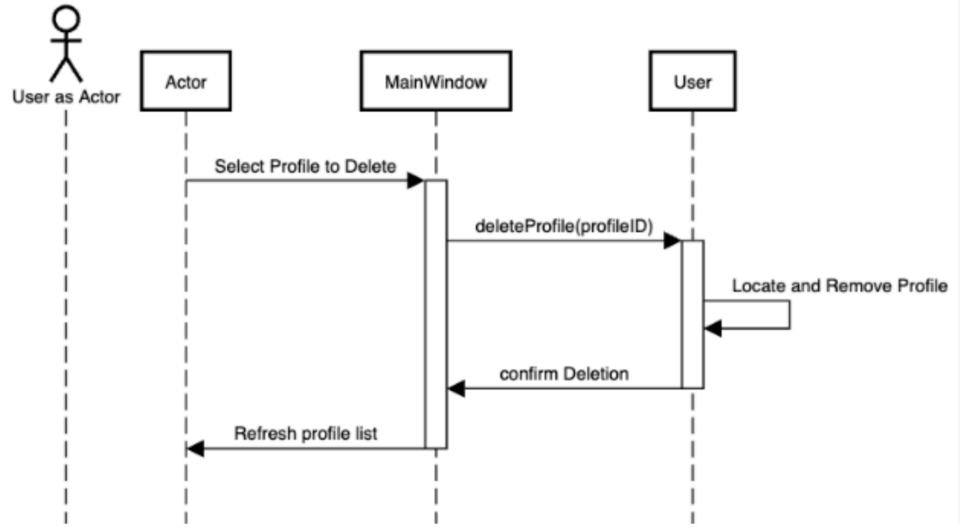
Editing User Profile



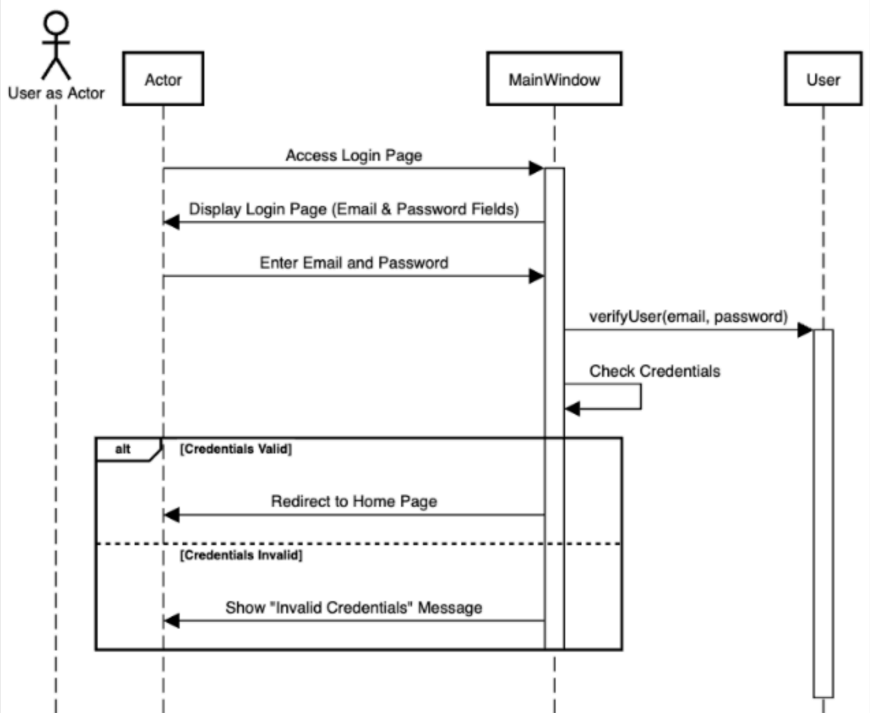
Adding a New Profile



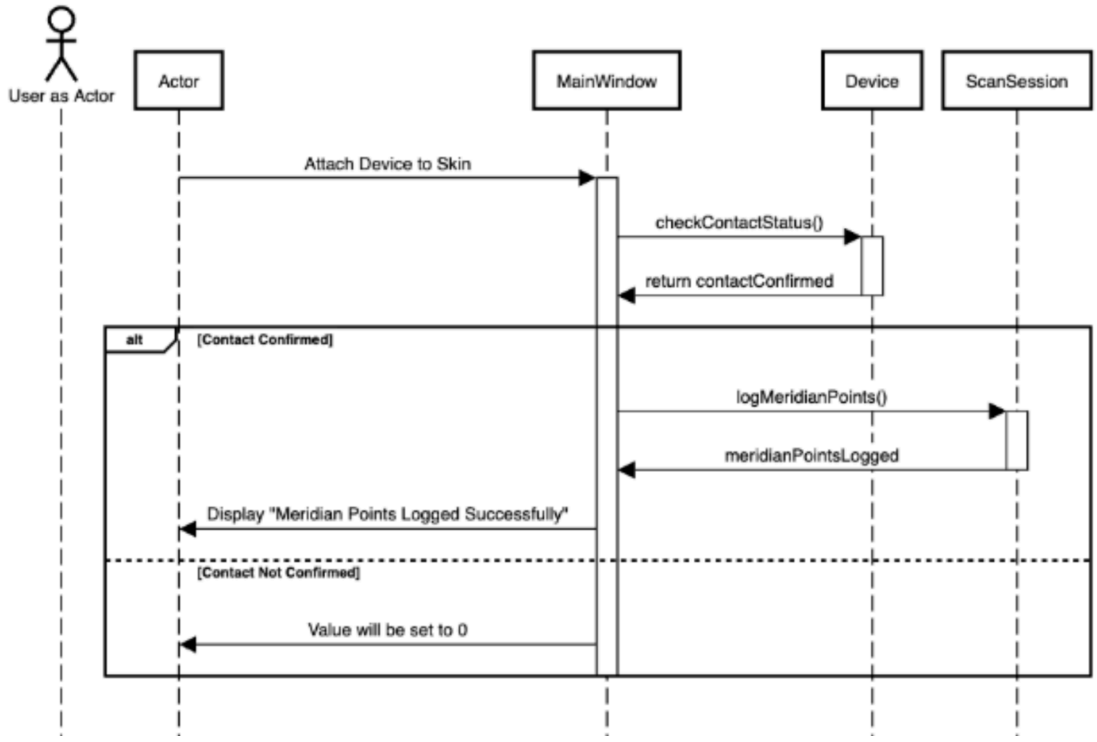
Deleting a New Profile



Logging into a User Account



Logging meridian Points bia Device Contact in RaDoTech Application



Traceability Matrix

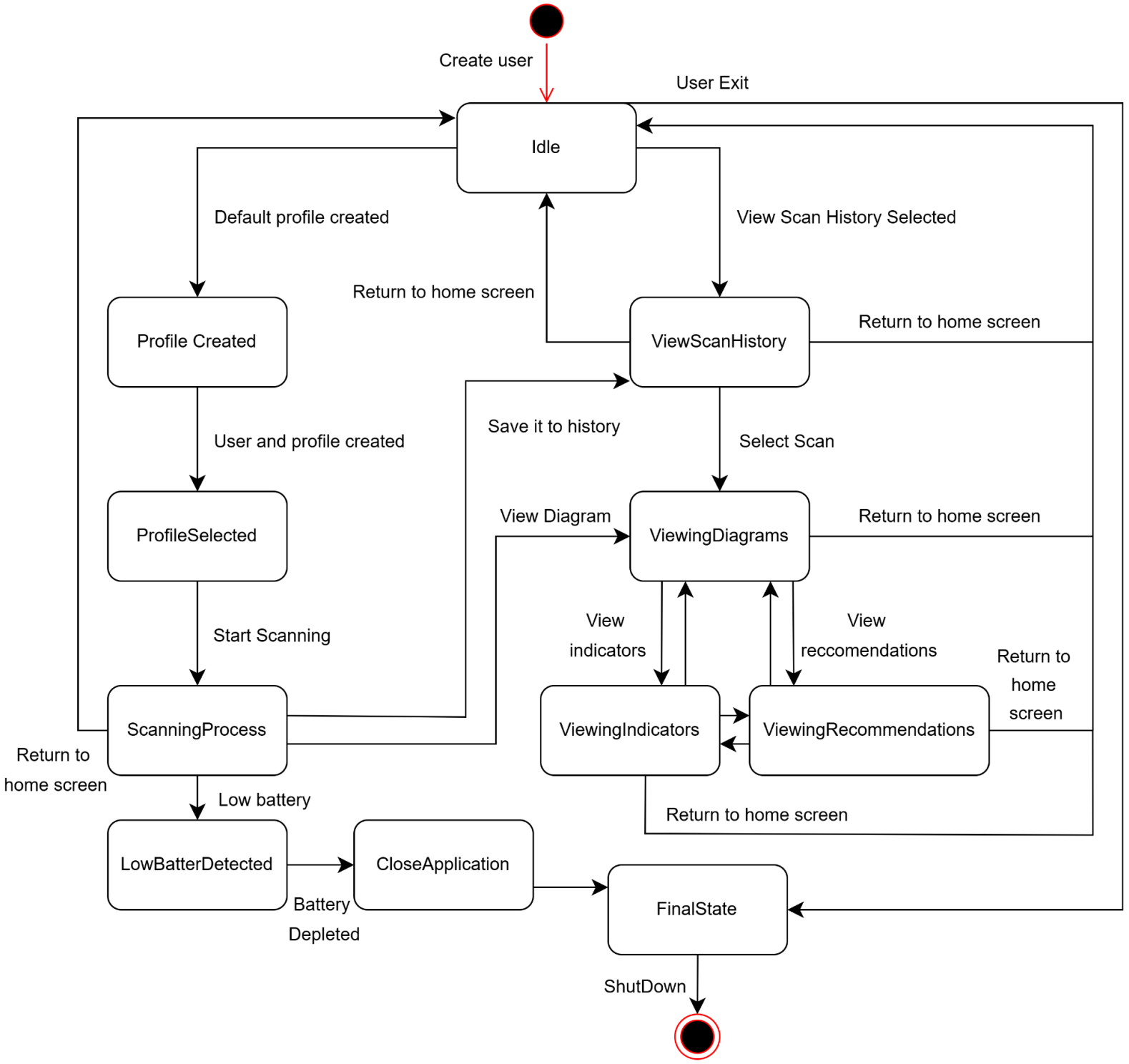
ID	Requirement	Related Use Case	Fulfilled By	Implemented by	Tested by	Description
1	A user/profile can be created.	UC1	User	saveUser()	Start the application, select the create profile option,and fill in the user info.Go to profile page from the menu and make sure the profile is created	User will create a user profile to use the application. A profile will be created for the user when the user is created
2	A user can hold multiple Profiles.	UC2	User, Profile	addProfile(const Profile& profile)	From the dashboard of a created user open up the side menu and go to the profile page. Select the new profile option and fill in the profile info. Repeat until the desired amount of profiles are created.	The device can hold up to 5 profiles.
3	Scanning process calculates metrics correctly based on profile information	UC2	Profile, ScanSession	calculateConductanceRange () setMeridianPoint(key, value) onFinishScanClicked()	Start the Scan session for the profile chosen and collect data from the user. Observe that all the values for the chart matches the conductance ranges according to the profile's age and weight	Process health metrics based off of conductance ranges of a profile based on their age and weight
4	Each profile can be updated.	UC3	User, Profile	updateProfile()	Go to the profiles page from the menu and select the profile that needs to be updated and put in the new info and save it. Select the updated profile to see the updated info	The profile information can be updated if needed.
5	Each profile can be deleted.	UC4	User, Profile	removeProfile(const QString& profileName)	Go to the profiles page from the menu and select the profile that needs to be deleted and select the delete profile option. Observe that the deleted profile does not remain	The profile can be deleted if it is not needed anymore.
6	Skin Contact	UC5	Mainwindow	handleSkinContact(); resetSkinContact();	Start the Scan Session, and observe the skin contact icon display whether the device is on skin or not.	Shows whether the device is in contact with the skin or not.

7	Store and allow users to access historical health data for trend analysis.	UC6	HistoricalData	saveCurrentSessionToHistory(ScanSession&)	Go to the history page from the menu and Observe that all the Scan Session done previously are saved.	Completed Scan sessions can be added to historical data that can be accessed later.

8	Data Processing and Visualization: Diagrams	UC7	Profile, Diagram	calculateBodyChartData();
				calculateBarChartData();
				colorDiagrams()
				adjustCharts()
					After the Scan Session has been completed. Observe that all the values for the body and chart diagrams are correct	Process raw data to generate health metrics and displays the calculated health metrics for the body and chart diagrams

9	Data Processing and Visualization: Indicators	UC8	Profile, Indicator	calculateEnergyLevel() calculateImmuneSystem() calculateMetabolism() calculatePsychoEmotional() calculateMusculoskeletal() CalculateLeft Meridian() CalculateRight Meridian() CalculateUpperMeridian() CalculateLowerMeridian() adjustStatus()	After the Scan Session has been completed. Observe that all the values for the Indicators are correct.	Process raw data to generate health metrics and displays the calculated health metrics for the body functioning indicators
10	Data Processing and Visualization: Recommendations	UC9	Profile, Recommendation	calculateAbnormities(double, double) getRecommendation()	After the Scan Session has been completed. Observe that all the recommendations printed match the data collected.	Process raw data to generate health metrics and displays the calculated health metrics for the recommendations
11	Empty battery powers off device	UC10	MainWindow, ScanSession	decreaseBattery()	Start the Scan Session, and observe the battery icon: battery level will lower as the treatment progresses, gracefully shut down the application when power levels reaches 0	Battery decreases with each scan and changes state based on power levels

RaDoTech State Machine Diagram



Design Patterns Explanation

One design pattern used in this application is the **Mediator Pattern** for behaviour. Behavior refers to how components interact and communicate with each other within the system.The MainWindow class serves as the central mediator, coordinating interactions between various components like Diagrams, Indicators, Recommendations, and User classes. Instead of these classes directly communicating with each other, they interact through the MainWindow, promoting loose coupling and simplifying interdependencies. This approach makes the system more modular, as changes to one component in one class do not heavily impact others. The Mediator Pattern encapsulates the complex logic of managing user actions, updates, and communication, making the MainWindow class responsible for facilitating interaction, data flow, and behaviour across the system.

Another design pattern used is the **Model-View-Controller (MVC)** for architecture. Architecture refers to the high-level organization of the system, where the application is divided into three distinct layers: Model, View, and Controller. The Indicators, Diagrams, and Recommendations classes represent the **Model**, containing the core data processing and logic. The MainWindow class acts as both the **View**, displaying data like charts and indicators, and the **Controller**, handling user inputs and coordinating updates between the interface and the **Model**.

The **Strategy Pattern** is used in Indicators and Diagrams to allow interchangeable algorithms for processing meridian point data, improving flexibility and reducing conditionals.
The combined use of these design patterns ensures the system remains scalable, maintainable, and easy to understand.