Use Case 1 - Conduct Health Measurement

Primary Actor(s): User, RaDoTech device
Stakeholders: User, RaDoTech Company
Pre-conditions: User has a charged RaDoTech device and the app installed.
Post-conditions: The user completed the body analysis with the RaDoTech device and the logs and user profiles are updated.
Main-success scenario:

1. User launches app
2. User selects their profile from a displayed list
3. User creates a new measurement/body analysis
4. App ensures device is connected
5. App requests user touches device to a meridian point
6. User touches device to specified point
7. Device closes electrical connection at point and sends reading to app
8. App stores reading to the current measurement
9. User removes device from skin
10. Repeat steps 5-9 for the other 23 points
11. App saves measurement as a record on the user's profile
12. App redirects user to the page regarding the newly created record
13. On this page, app renders bar graph of data
14. On this page, app displays text summary and recommendation of future action
15. User turns off the app.

Extensions:

2a. User does not currently have a profile
    2a1. User selects to create a new profile
    2a2. User inputs personal information such as name, age
    2a3. App saves profile to device
    2a4. Scenario resumes at step 2
4a. Device failed to connect
    4a1. User is notified that the connection failed.
    4a2. User tries to connect the device again.
4b. Device has no battery left.
    4b1. User has to charge the device to continue.
6a. The device is not attached to skin
    6a1. User is prompted to attach the device to the skin.
11a. User wants to view history
    11a1. User can view the history of data collected in the history tab.
13a. User wants additional health insights
    13a1. User can view additional health insights for the latest body scan.

Use Case 2 - View History

Primary Actor(s): User, RaDoTech App
Stakeholders: User, RadoTech
Pre-conditions:

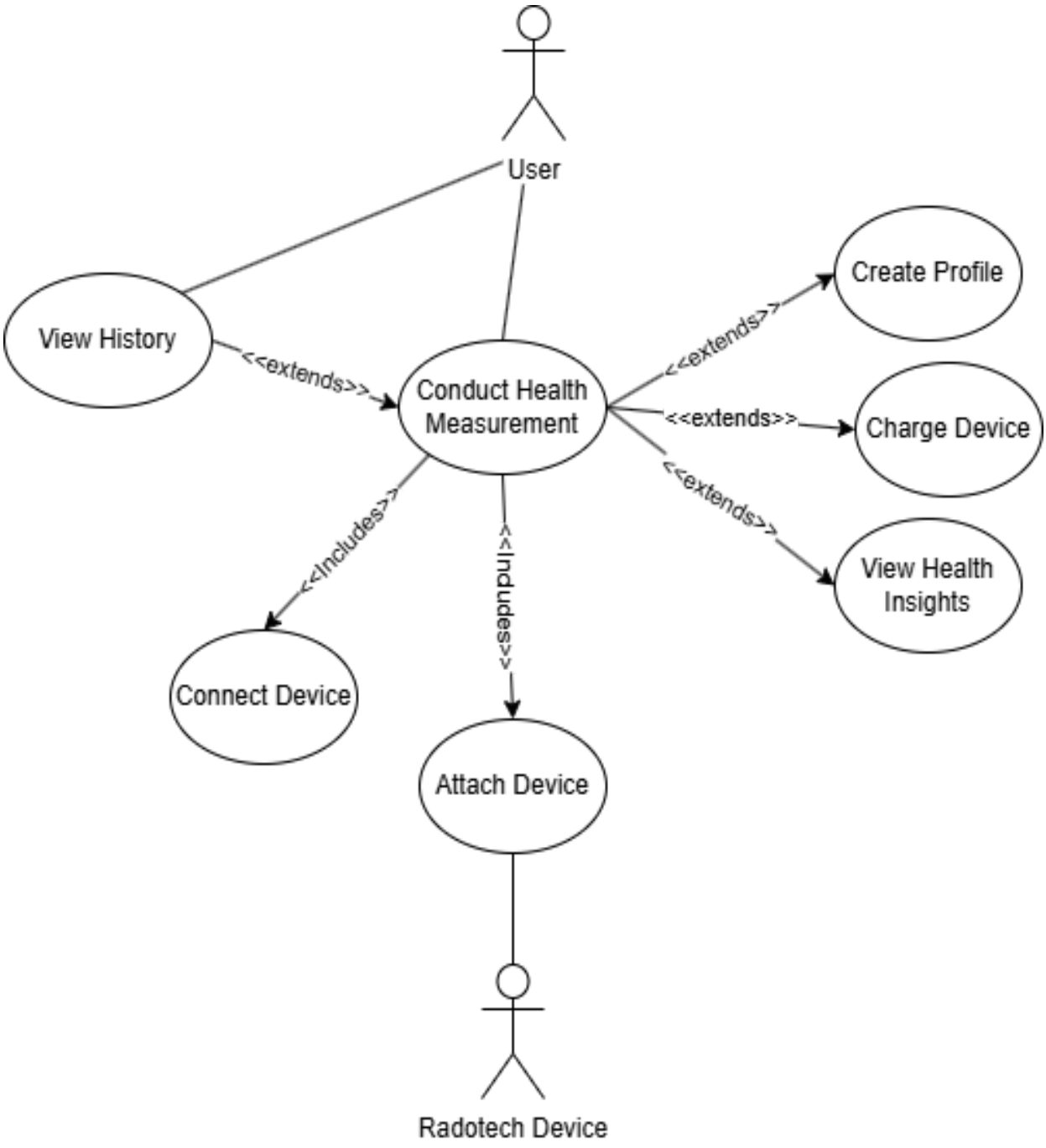
- User has a RaDoTech device
- User has RaDoTech app installed on their phone
- User has at least one profile created in the app
- User has conducted previous health scan(s)

Post-conditions: User can view past health scans for their specific user profile
Main-success scenario:

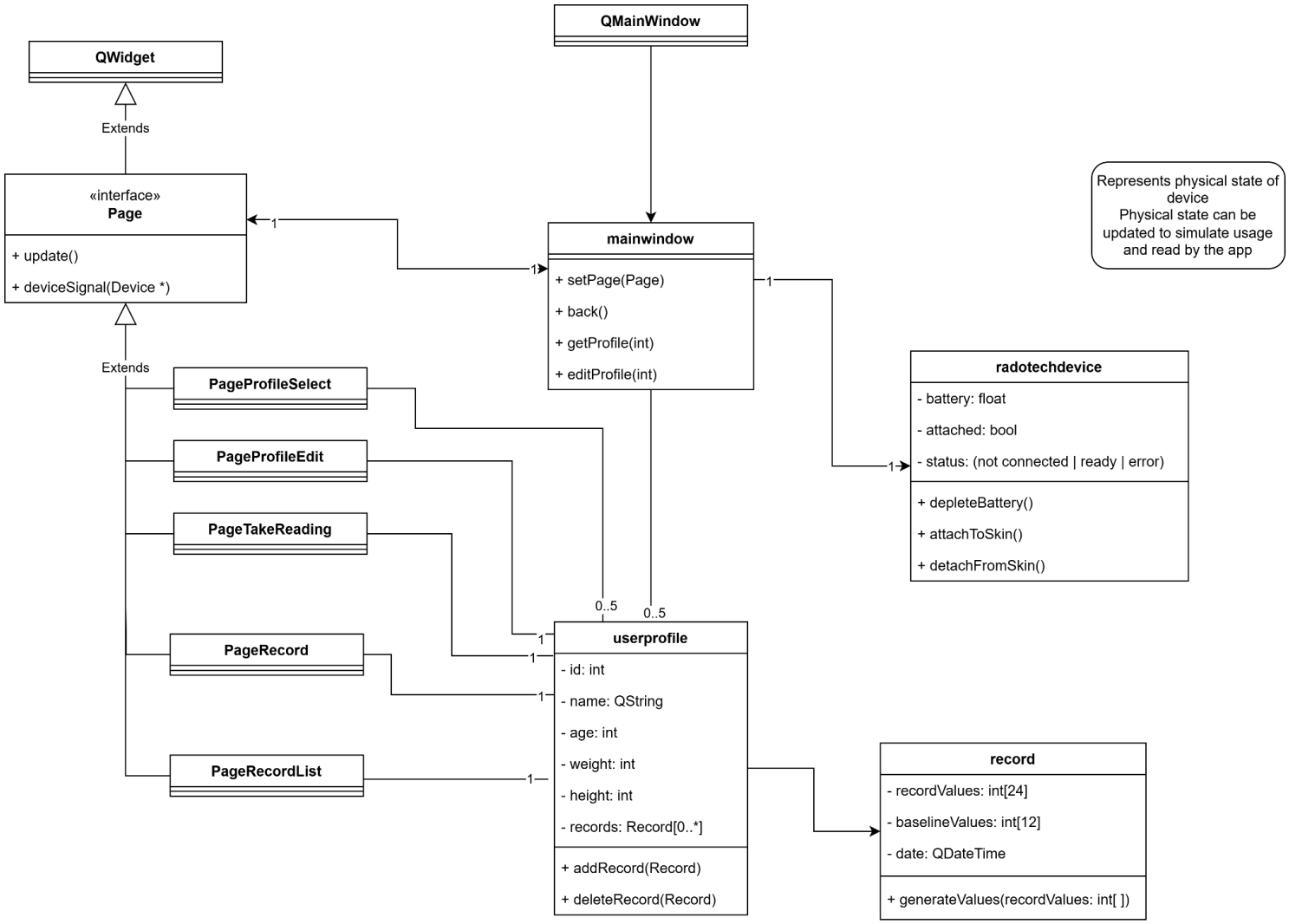
1. User selects a profile
2. Health scan history is selected by the user
3. App displays history of health scans taken
4. User selects a health scan record
5. App displays information relating to data patterns from health scan
6. App displays recommendations from healthcare provider based on latest record
7. After analysis of health scan, app will suggest specific RaDoTech supplements that the user is lacking (e.g. CardioTune if blood pressure is low)

Extensions:
2a. User has not conducted any previous health scans
    2a1. User conducts a health scan

Use case diagram:



UML Class Diagram:



Program Design:

The application is designed with modularity, scalability and user-centric interaction in mind, leveraging Qt and essential design patterns:

1. Class Structure
    a. UserProfile and Record manage user-specific data and health measurements. RadotechDevice encapsulates device logic, including states (`READY`, `NOT_CONNECTED`, `ERROR`).
    b. **Justification:** This modular design ensures easy scalability and clean separation of concerns

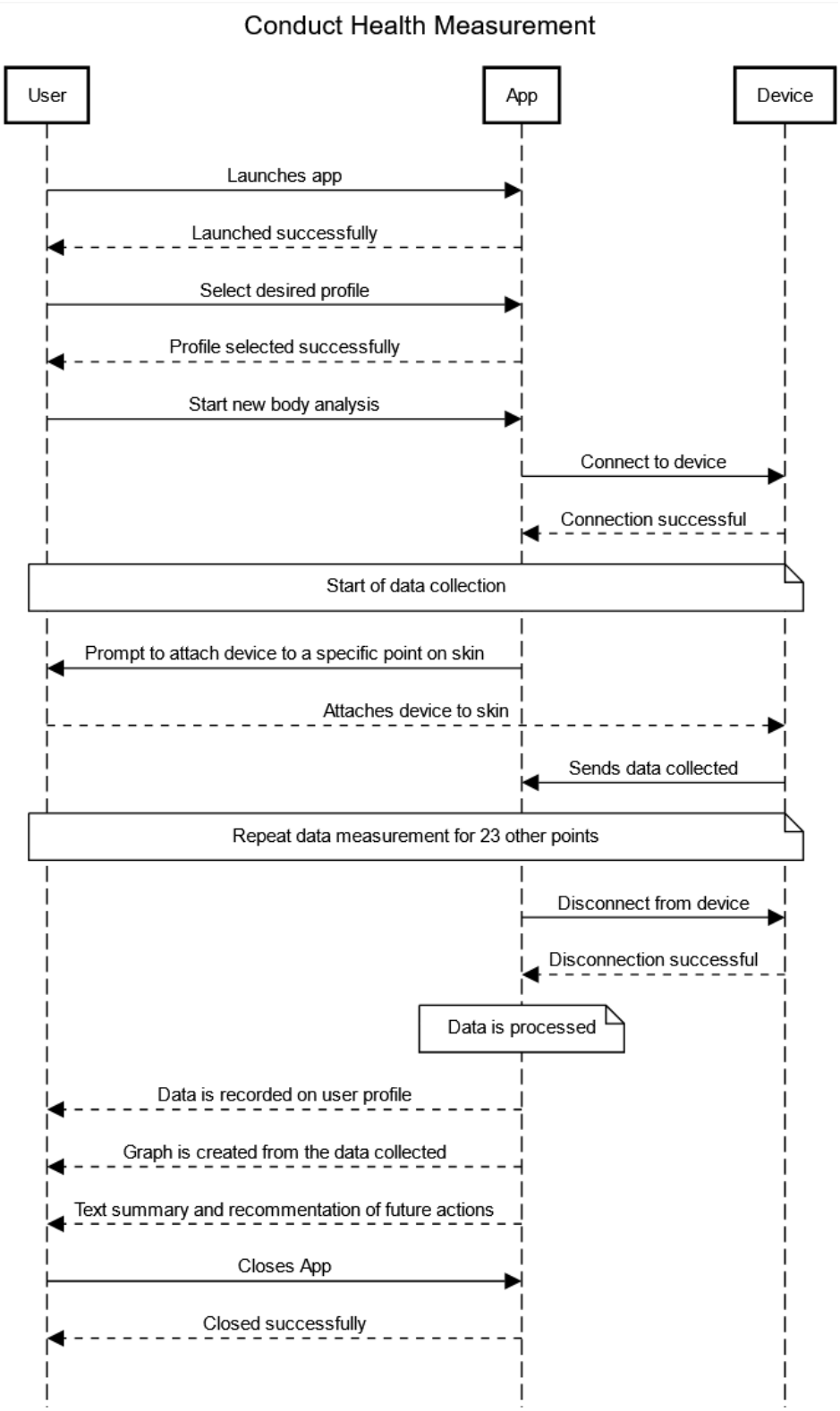
2. Key Design Patterns
    a. **Observer (Qt):** Signals and slots dynamically update UI components like `deviceSignal()` in PageTakeReading for state changes.
    b. **Mediator:** MainWindow manages navigation between UI pages to centralize control.
    c. **State:** RadotechDevice enforces operations based on connection and battery state.
    d. **Justification:** These patterns ensure decoupled, extensible and robust interactions.

3. Realizing Use Cases
    a. **Health Measurement (Use Case 1):** The State pattern ensures proper device readiness, with `nextStatus()` guiding the measurement workflow.
    b. **History Viewing (Use Case 2):** PageRecordList retrieves and displays user records seamlessly.

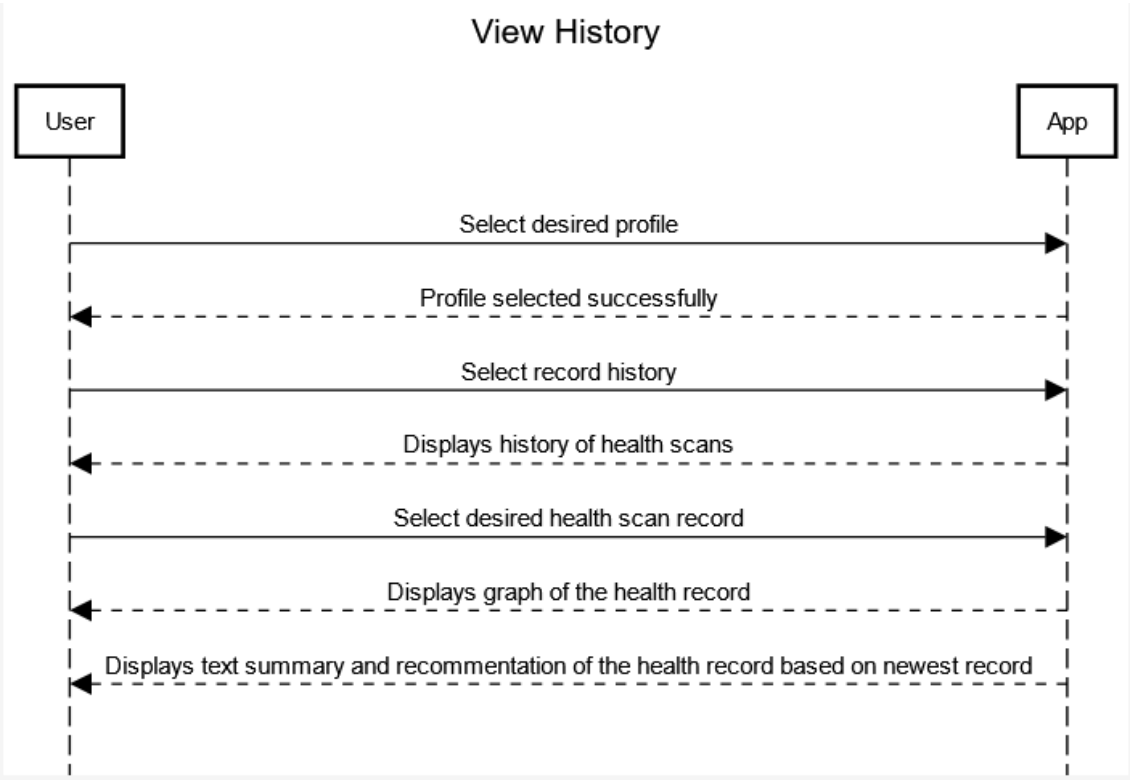
4. Database and Testing
    a. SQLite persists user data with modular functions for clean SQL handling.
    b. `generateValues()` in Record uses the Strategy pattern to simulate test data.
    c. **Justification:** Supports real-world scalability and proper debugging.

Sequence Diagrams:

Use Case 1:

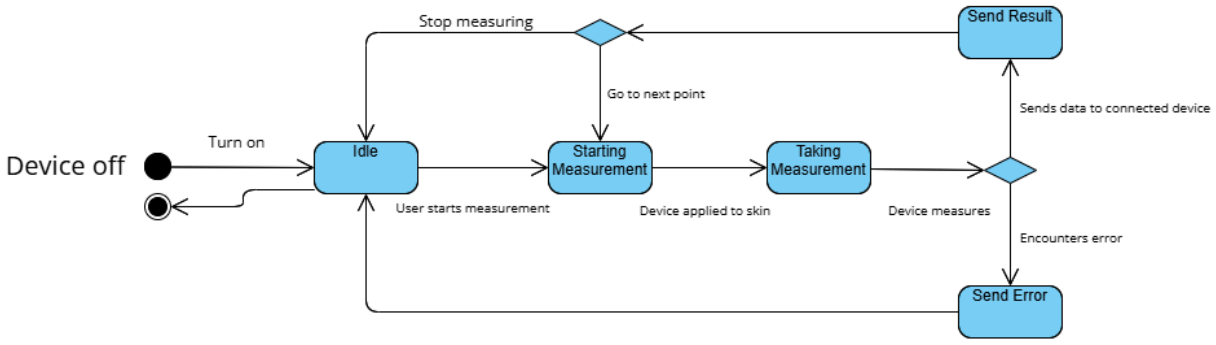


Use Case 2:

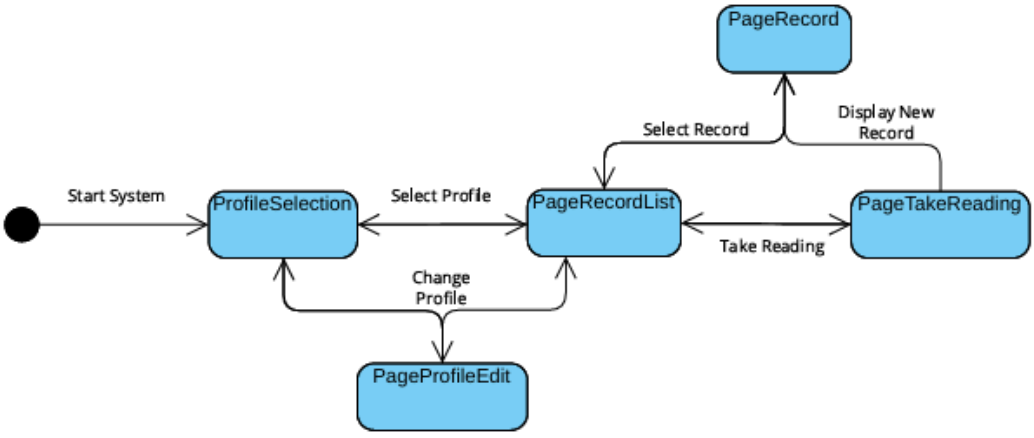

View History
User
App
Select desired profile
Profile selected successfully
Select record history
Displays history of health scans
Select desired health scan record
Displays graph of the health record
Displays text summary and recommentation of the health record based on newest record

State Machine Diagrams:

Radotech Device



MainWindow



Requirements Traceability Matrix:

ID	Requirements	Related Use Case	Description/ Design Elements	Implemented By	Tested By
1	The app must allow users to conduct health measurements using the RaDoTech device.	Use Case 1	Use Case 1, Sequence Diagram 1, State Machine Diagram Sequence Diagram 1 shows the interaction between the user and the app for measurements. State Machine Diagram details device states.	pagetakereading.cpp, record.cpp	Verify health measurement workflow with all 24 points.
2	The app must allow users to view the history of health measurements.	Use Case 2	Use Case 2, Sequence Diagram 2 Sequence Diagram 2 outlines history retrieval and display flow.	pagerecordlist.cpp, pagerecord.cpp	Verify the correct display of past records in history.
3	The app must provide a profile management system for user-specific data.	Use Case 1, Use Case 2	Class Diagram (UserProfile, Record), PageProfileEdit Class Diagram links UserProfile and Record classes to profile management features. PageProfileEdit enables user edits.	userprofile.cpp, pageprofileedit.cpp	Validate profile creation, update, and deletion functionalities.

4	The app must ensure connectivity and check the battery status of the RaDoTech device.	Use Case 1	Class Diagram (RadotechDevice), State Machine Diagram Class Diagram for RadotechDevice includes connectivity and battery state logic. State Machine Diagram covers state transitions.	radotech.cpp, mainwindow.cpp	Test device connection and battery checks with valid/invalid scenarios.
5	The app must generate and display recommendations based on health measurements.	Use Case 1	PageTakeReading, PageRecord, Sequence Diagram 1 Sequence Diagram 1 and PageRecord support recommendation display based on recorded values.	pagetakereading.cpp, pagerecord.cpp	Ensure recommendations are displayed based on measurement values.