COMP 3004 Final Project

t:slim Insulin Pump Use Cases

Members:
Shakil Muhammad (101259661)
Pavle Vujicic (101159343)
Jacob Rada (101265520)
Shivam Khanna (101268691)

Content:

Use Case 1: Creating a profile

Description: This use case describes how a user creates a personal insulin delivery profile on the insulin pump.

Actors: Users

Pre-condition: The insulin pump is turned ON, and the user accesses the personal profile settings.

Post-condition: A new insulin delivery profile is successfully created and saved.

Main Sequence:

1. The user navigates to the "Personal Profiles" section.
2. The system displays existing profiles and an option to create a new one.
3. The user selects "Create New Profile."
4. The system prompts the user to enter a profile name (eg “Morning Routine” or “Exercise Mode”).
5. The user enters basal rates, carbohydrate ratios, correction factors, and target glucose levels.
6. The user reviews and confirms the settings.
7. The system saves the new profile and displays a confirmation message.

Extensions:

4a. If the user does not enter a name, the system prompts them to do so before proceeding

Use Case 2: Insulin pump setup

Description: This use case explains how users set up and power their t:slim X2 insulin pump.

Actors: Users

Pre-condition: The insulin pump is not yet configured or powered ON.

Post-condition: The insulin pump is set up and ready for use.

Main Sequence:

1. The user charges the pump using the provided USB cable.
2. The battery indicator displays charging progress.
3. The user presses and holds the power button to turn on the pump.
4. The startup sequence completes, and the main screen is displayed.
5. The user navigates the touchscreen interface to access initial setup options.
6. The user configures a PIN-based lock screen for security.

Use Case 3: Interacting with insulin pump

Description: This use case describes how users interact with the primary interface of the insulin pump.

Actors: Users

Pre-condition: The insulin pump is powered ON.

Post-condition: The user understands the interface and navigation of the pump.

Main Sequence:

1. The user accesses the home screen of the insulin pump.
2. The device displays key indicators, including insulin fill level, battery life, and CGM data.
3. The user reviews the "Insulin on Board" (IOB) indicator for active insulin tracking.
4. The user interacts with navigation buttons such as the bolus calculator or system settings.
5. The user returns to the home screen by tapping the Tandem logo.

Extensions:

3a. If the user selects the bolus button, the system navigates to the bolus calculator.

3b. If the user selects the options button, the system displays insulin settings and alerts.

Use Case 4: Starting Insulin Delivery

Name: Starting Insulin Delivery

Primary Actor(s): User

Stakeholders:

User- Initiates and monitors insulin delivery.

Healthcare Provider- Relies on accurate insulin delivery for patient care.

Control IQ Technology- Monitors glucose levels and adjusts insulin delivery.

Pre-condition(s):

- The pump is powered on and unlocked.
- A personal profile with basal rates is configured.
- The pump is not currently delivering insulin.

Success guarantee(s) (Post-conditions):

- Insulin delivery is active.
- The current status screen displays the active basal rate.
- The system logs the start of insulin delivery.

Main success scenario:

1. The user navigates to the "Basal Rate" screen via the options menu.
2. The user selects a basal rate from their active personal profile or manually configures a new rate.
3. The user confirms the selected basal rate.
4. The pump begins delivering insulin at the specified rate.
5. The system logs the start of insulin delivery.

Extensions:

1a. Invalid Basal Rate Configuration

1a1. The user inputs an invalid basal rate (e.g., negative value). The system displays an error and prompts for correction.

4a. Low Glucose Detected

4a1. Control IQ technology detects low glucose levels (below 3.9 mmol/L). The system automatically suspends insulin delivery and logs the event.

Use Case 5: Stopping Insulin Delivery

Name: Stopping Insulin Delivery
Primary Actor(s): User or Control IQ Technology
Stakeholders:
User: Stops insulin delivery manually or monitors automatic stops.
Healthcare Provider: Relies on accurate insulin delivery logs for patient care.
Control IQ Technology: Automatically stops insulin delivery based on glucose levels.
Pre-condition(s):

- The pump is powered on and unlocked.
- Insulin delivery is currently active.

Success guarantee(s) (Post-conditions):

- Insulin delivery is stopped.
- The current status screen reflects the stopped state.
- The system logs the stop event.

Main success scenario:

1. Manual Stop

1a. The user navigates to the "Basal Rate" screen.
1b. The user selects the option to stop insulin delivery.
1c. The pump stops delivering insulin.
1d. The system logs the stop event.

2. Automatic Stop (Control IQ)

2a. Control IQ technology detects low glucose levels (below 3.9 mmol/L).
2b. The pump automatically suspends insulin delivery.
2c. The system logs the suspension event and displays an alert to the user.
Extensions:
1c. Pump Fails to Stop Delivery
1c1. The pump fails to stop insulin delivery. The system displays an error message and logs the failure.

Use Case 6: Resuming Insulin Delivery

Name: Resuming Insulin Delivery

Primary Actor(s): User

Stakeholders:

User: Resumes insulin delivery after it has been stopped.

Healthcare Provider: Relies on accurate insulin delivery logs for patient care.

Pre-condition(s):

- The pump is powered on and unlocked.
- Insulin delivery was previously stopped (manually or automatically).

Success guarantee(s) (Post-conditions):

- Insulin delivery is active.
- The current status screen reflects the resumed state.
- The system logs the resumption of insulin delivery.

Main success scenario:

1. The user navigates to the "Basal Rate" screen.
2. The user selects the option to resume insulin delivery.
3. The pump resumes delivering insulin at the previously active basal rate or a newly configured rate.
4. The system logs the resumption of insulin delivery.

Extensions:

3a. Pump Fails to Resume Delivery

3a1. The pump fails to resume insulin delivery. The system displays an error message and logs the failure.

Use Case 7: Displaying Pump Info

Primary Actor: Users
Scope: t:slim X2 device lifetime.
Level: User-goal
Stakeholders and Interests: none
Precondition:
Minimal Guarantee: Recent information is logged.
Success Guarantee: User is able to view current and recent health information.
Main Success Scenario:

1. User goes about their day, using the device as normal
2. Machine collects general information and stores in short term memory
3. User accesses Current Status screen from
4. Recent general data is displayed.

Extensions:
2a) Data collected includes: time and amount of the last bolus, changes in basal rates, or alerts triggered by CGM readings

Use Case 8: Displaying Pump History

Primary Actor: Users
Scope: t:slim X2 device lifetime.
Level: User-goal
Stakeholders and Interests: Healthcare providers
Precondition:
Minimal Guarantee: Significant action details are saved.
Success Guarantee: User is able to view a usage history of at least 90 days.
Main Success Scenario:

1. User uses device as normal for an extended period of time
2. Machine collects and saves data to long term memory
3. User accesses the Pump History screen.
4. Data is displayed to the user.
5. Users or healthcare providers may examine data.

Extensions:
2a) Information such as total insulin delivery by basal and bolus types into units and percentages. Such is stored by the day. Other events such as Alerts are also stored.
2b) Once the maximum number of events is reached, oldest events are erased from memory and replaced with new events.
4a) Data can be displayed in a variety of days, e.g. viewing insulin delivery by days, weeks, months etc…, filtering events by type

Use Case 9: Delivering Insulin through Manual Bolus

Primary Actor: Users
Scope: t:slim X2 daily use.
Level: User-goal
Stakeholders and Interests: none
Precondition: Device is set up and attached to body, and is turned on
Minimal Guarantee: Process can be aborted by the user anytime
Success Guarantee: Specified amount of insulin is delivered to the body through the bolus based on the user's request.
Main Success Scenario:

1. User presses the Bolus button on the user interface home screen or device and opens the Bolus Calculator Screen
2. Blood glucose level and carbohydrate intake are automatically obtained from the CGM
3. An appropriate dose is calculated based on programmed settings
4. User presses start button and insulin dose is administered
5. Status on home page is updated accordingly

Extensions:

2a) User is able to manually enter specified values for blood glucose level and carbohydrate intake

3a) User is able to override the dose suggested by the machine.

4a) User may decide on a time period for delivery, should they want delivery over a long period, or an immediate bolus

4b) Delivery process may be cancelled at any time.

Use Case 10: Handling Low Battery Alert

Name: Handling Low Battery Alert
Primary Actor(s): User
Stakeholders:
User: Responds to low battery alerts.
Healthcare Provider: Relies on uninterrupted insulin delivery for patient care.
Pre-condition(s):

- The pump is powered on and unlocked.
- The battery level is critically low.

Success guarantee(s) (Post-conditions):

- The pump is charging.
- The low battery alert is cleared once the battery level is sufficient.
- The system logs the low battery event and charging status.

Main success scenario:

1. The pump detects a low battery level and displays an alert on the screen.
2. The user connects the pump to a power source using the provided USB cable.
3. The pump begins charging, and the battery indicator updates in real-time.
4. The system logs the low battery event and charging status.

Extensions:
2a. Pump Not Connected to Power Source
2a1. The pump is not connected to a power source. The alert persists, and the pump may shut down.

Use Case 11: Handling Occlusion Alert

Name: Handling Occlusion Alert

Primary Actor(s): User

Stakeholders:

User: Responds to occlusion alerts.

Healthcare Provider: Relies on accurate insulin delivery logs for patient care.

Pre-condition(s):

- The pump is powered on and unlocked.
- An occlusion (blockage) is detected in the infusion set.

Success guarantee(s) (Post-conditions):

- The occlusion alert is cleared.
- Insulin delivery resumes normally.
- The system logs the occlusion event and resolution.

Main success scenario:

1. The pump detects an occlusion and displays an alert on the screen.
2. The user checks the infusion site for blockages or kinks in the tubing.
3. The user resolves the issue (e.g., replaces the infusion set).
4. The pump resumes normal operation.
5. The system logs the occlusion event and resolution.

Extensions:

3a. Issue Not Resolved

3a1. The issue is not resolved. The alert persists, and insulin delivery remains suspended.

Use Case 12: Handling CGM Disconnection Alert

Name: Handling CGM Disconnection Alert

Primary Actor(s): User

Stakeholders:

User: Responds to CGM disconnection alerts.

Healthcare Provider: Relies on accurate glucose data for patient care.

Pre-condition(s):

- The pump is powered on and unlocked.
- The CGM is disconnected or not transmitting data.

Success guarantee(s) (Post-conditions):

- The CGM disconnection alert is cleared.
- The pump resumes receiving CGM data.
- The system logs the disconnection event and resolution.

Main success scenario:

1. The pump detects a CGM disconnection and displays an alert on the screen.
2. The user checks the CGM device and connection.
3. The user reconnects or troubleshoots the CGM device.
4. The pump resumes receiving CGM data.
5. The system logs the disconnection event and resolution.

Extensions:

3a. CGM Remains Disconnected

3a1. The CGM remains disconnected. The alert persists, and Control IQ technology may not function optimally.

Use Case 13: Handling Pump Shutdown

Name: Handling Pump Shutdown

Primary Actor(s): User

Stakeholders:

User: Responds to critical pump shutdowns.

Healthcare Provider: Relies on accurate insulin delivery logs for patient care.

Pre-condition(s):

- The pump is powered on and unlocked.
- A critical failure (e.g., hardware malfunction) occurs.

Success guarantee(s) (Post-conditions):

- The pump is either restarted or remains shut down.
- The system logs the shutdown event and restart attempts.

Main success scenario:

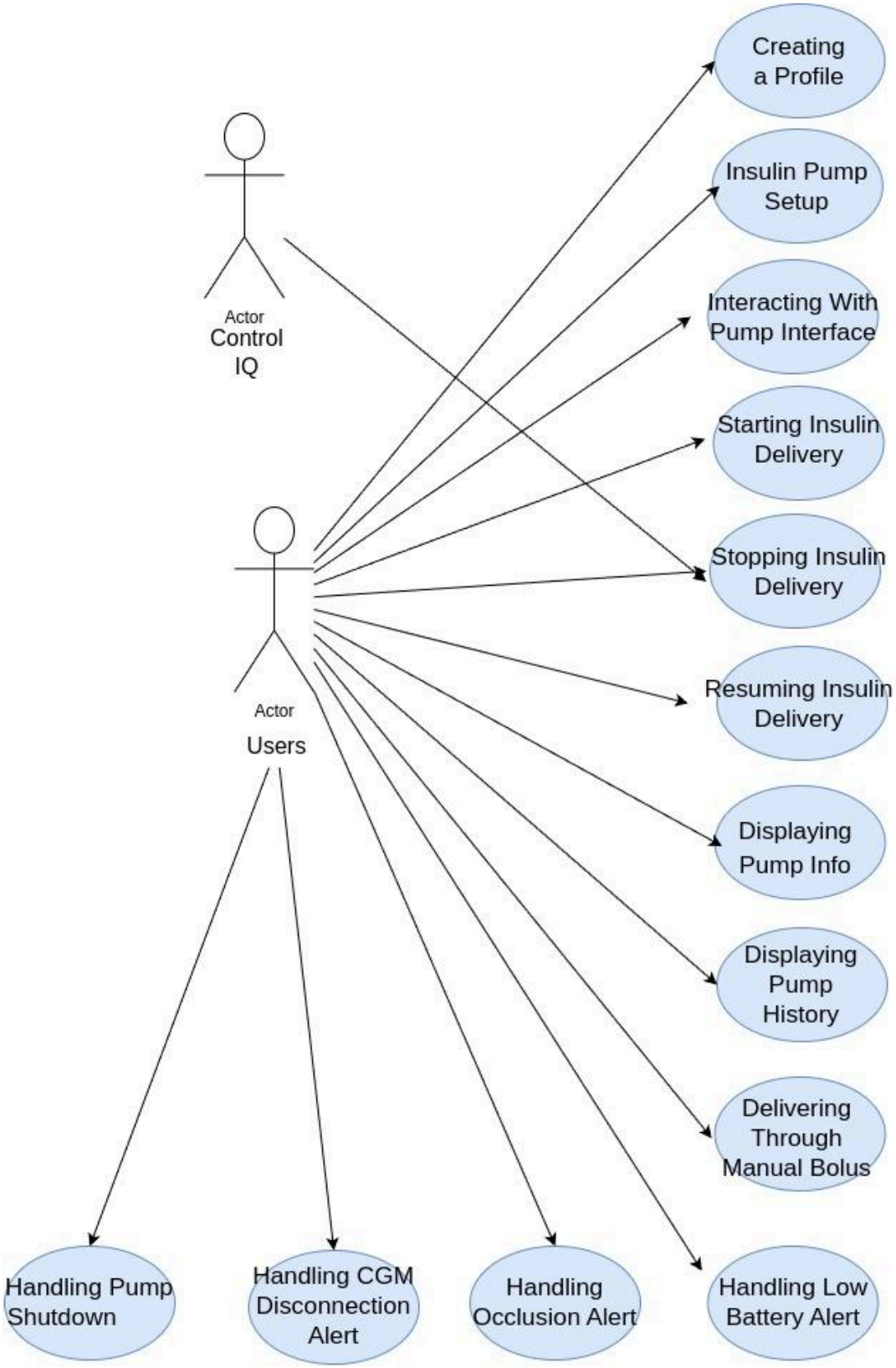
1. The pump detects a critical failure and shuts down.
2. The system displays a shutdown message and guidance for restarting or contacting support.
3. The user attempts to restart the pump.
4. If the restart is successful, the pump resumes normal operation.
5. If the restart fails, the user contacts technical support.
6. The system logs the shutdown event and restart attempts.

Extensions:

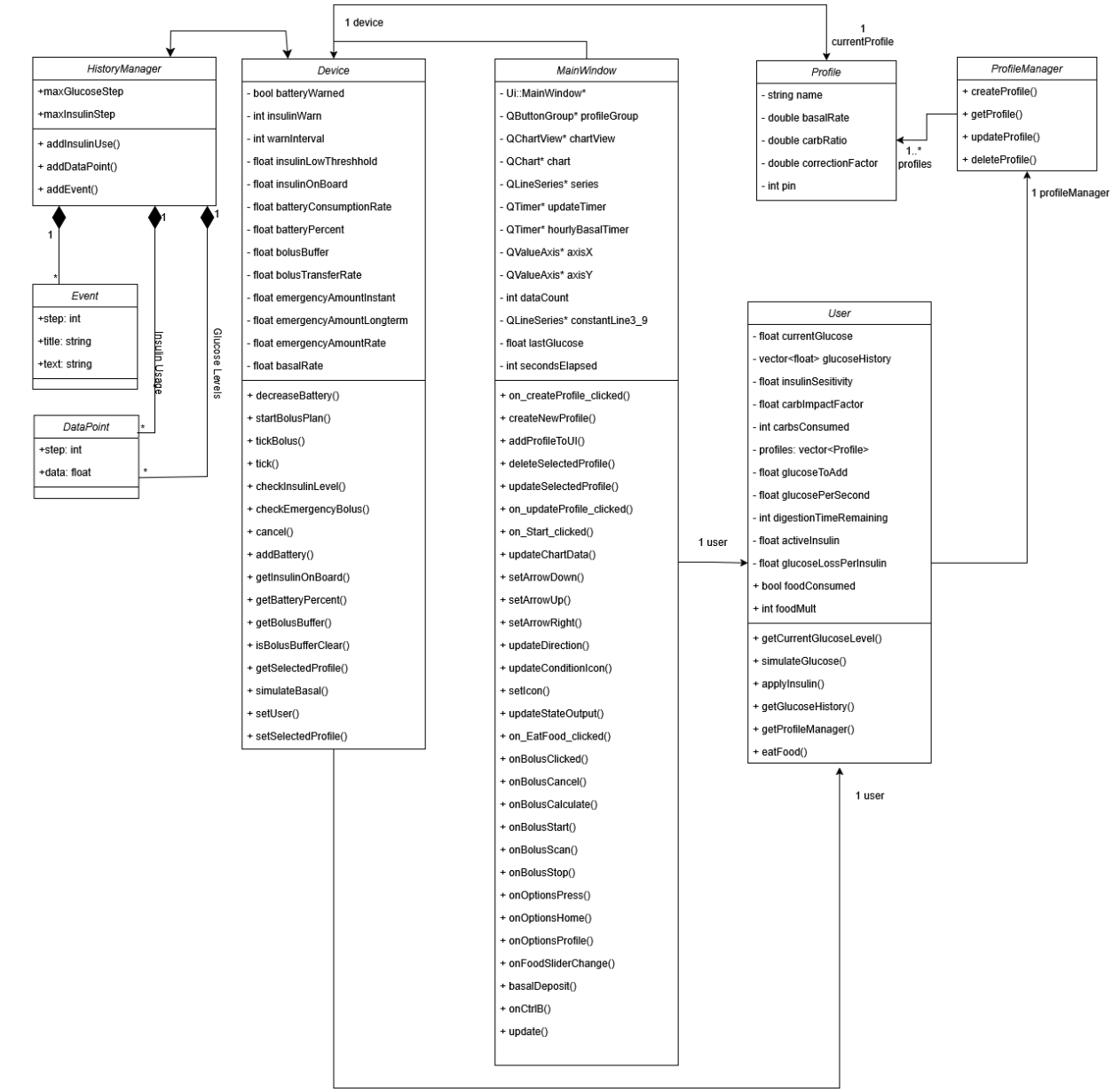
4a. Pump Cannot Be Restarted

4a1. The pump cannot be restarted. The user must switch to manual insulin delivery methods.

Use Case Diagram



UML Class Diagram

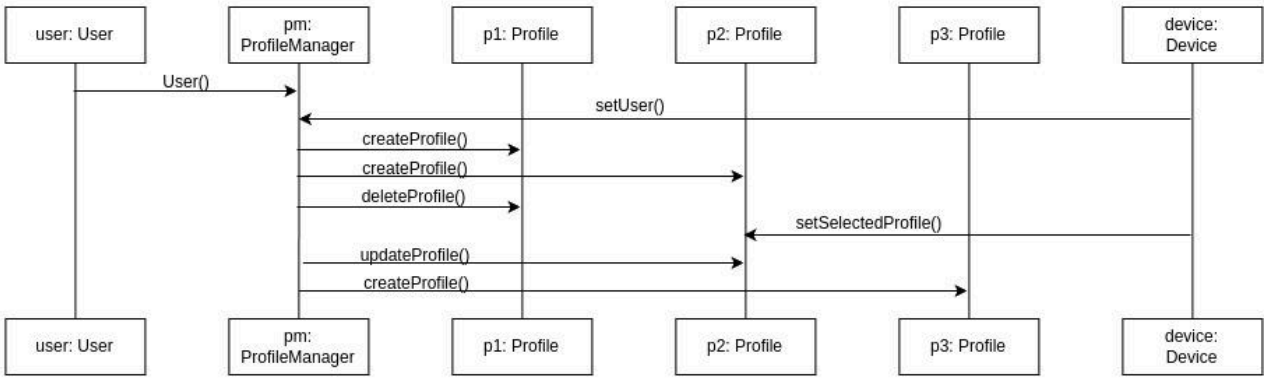


UML Sequence Diagrams:

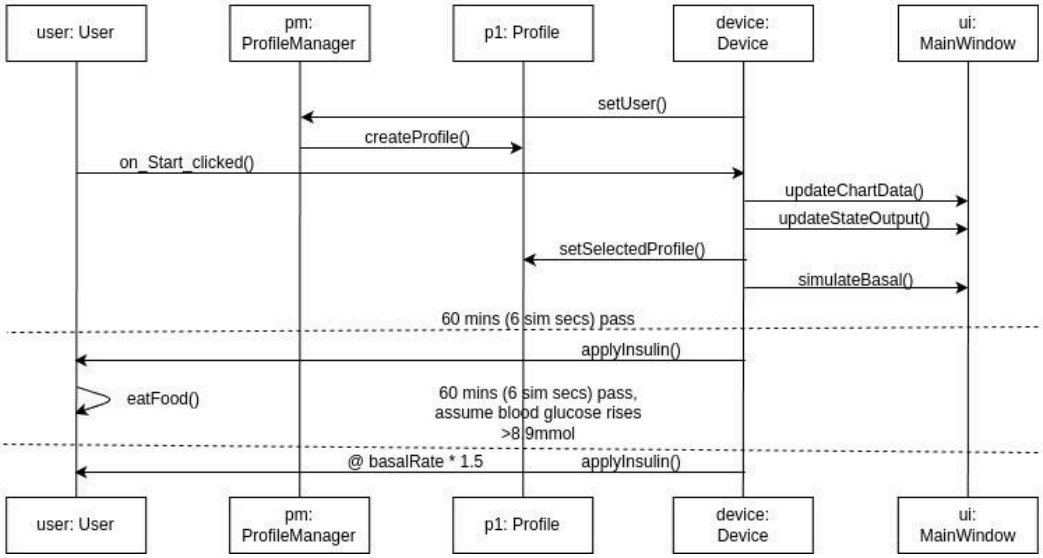
Success Scenarios-

Success Scenario #1: Profile Management

(Start with 2, delete 1, read remaining profile and update, create one more)



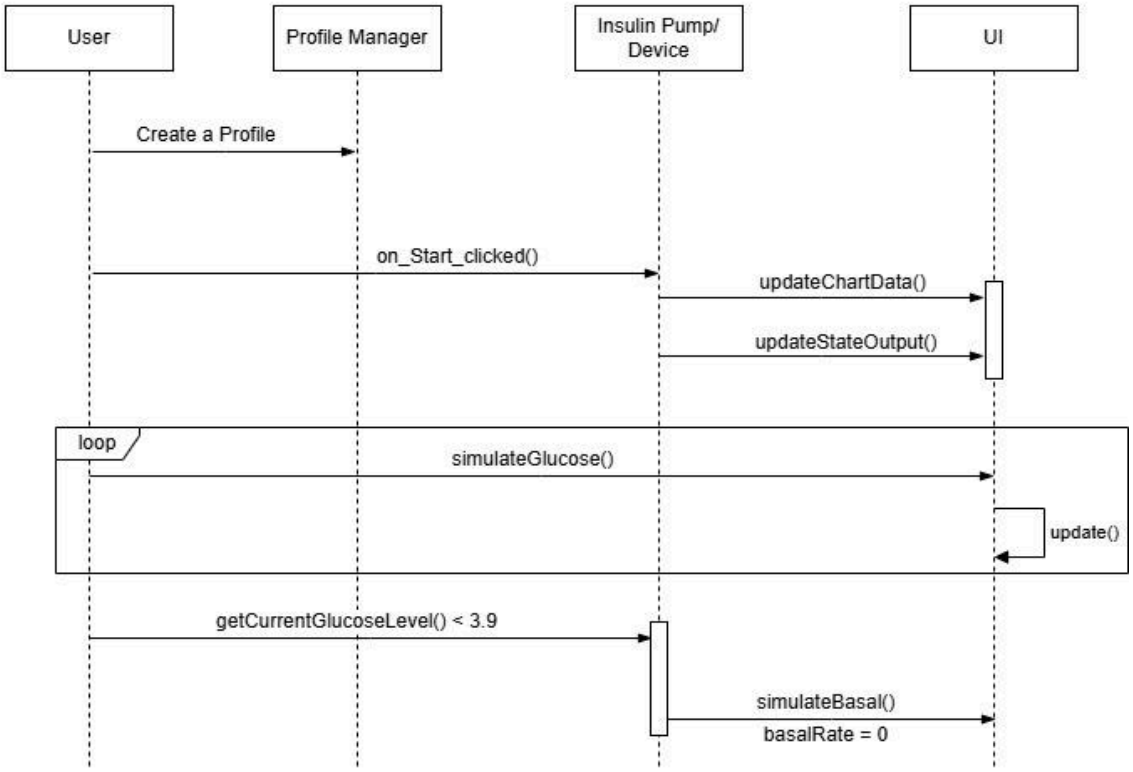
Success Scenario #2: Deposit Hourly Basal



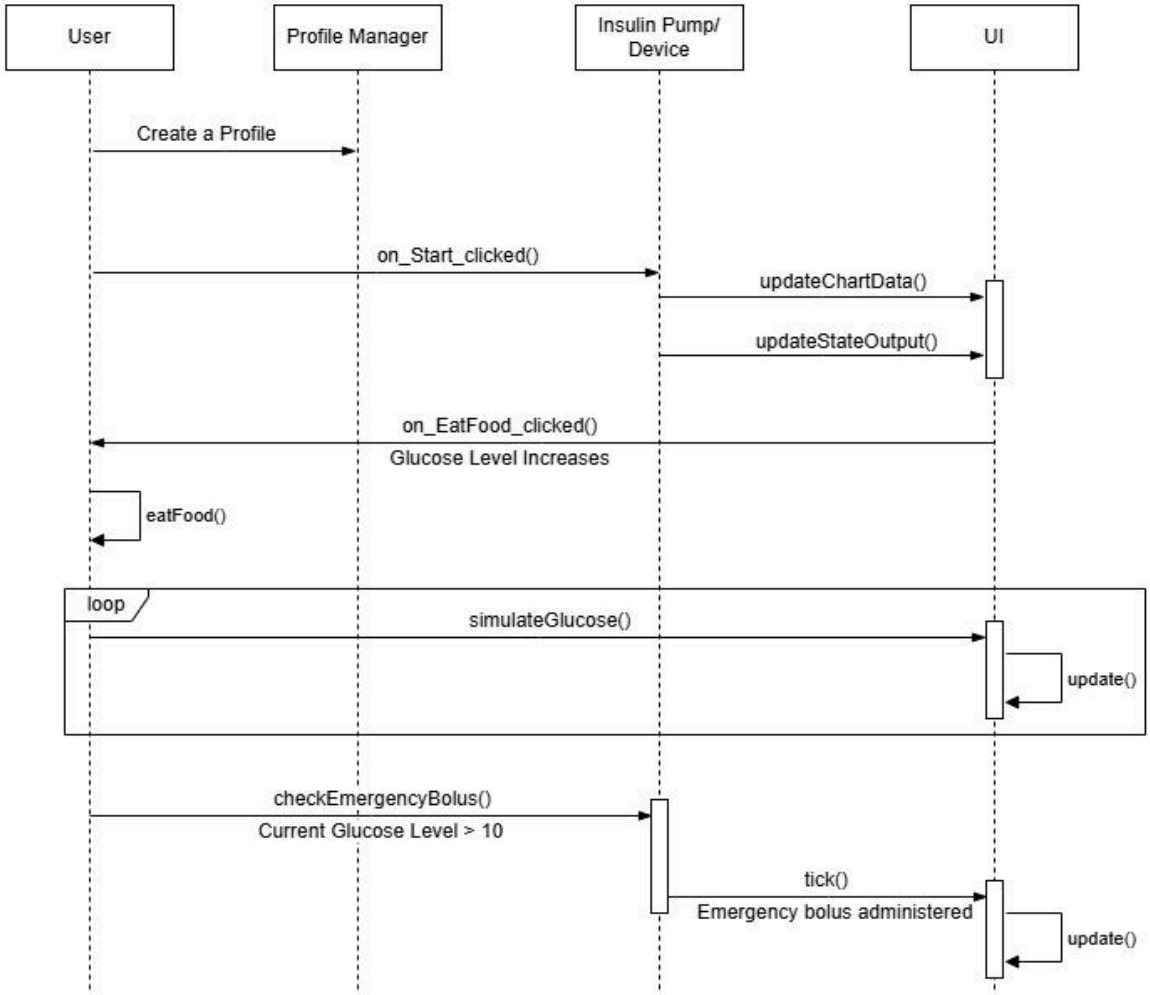
Safety Scenarios-

(next page->)

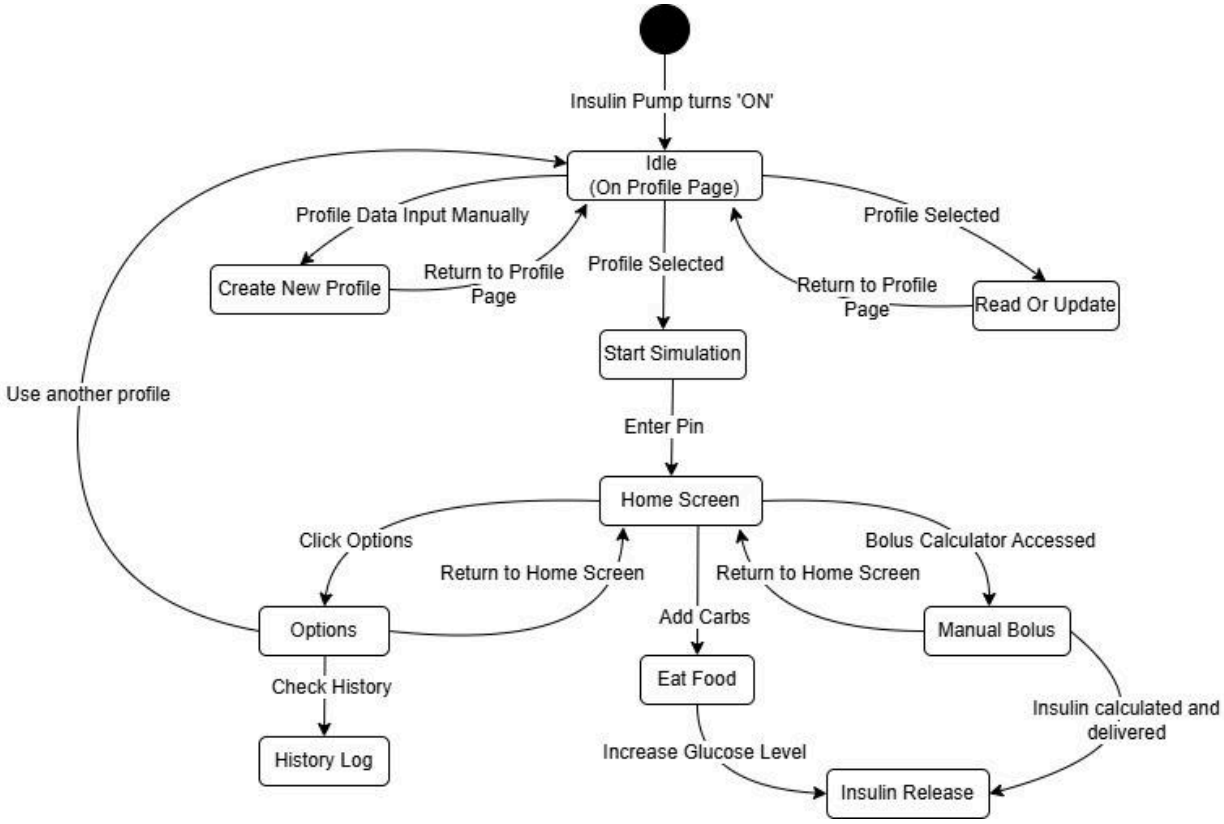
Safety Scenario #1: Low Blood Glucose



Safety Scenario #2: Deposit Emergency Bolus



UML State Machine Diagram:



Traceability Matrix:

ID	Requirement	Related Use Case	Implemented - By	Tested - By	Description
1	Implement primary interface	Interacting with Pump Interface (UC3)	MainWindow, MainWindow.ui, Device	Run the interface simulation in Qt; observe UI elements and interactions.	Replicates the pump's home screen where users can monitor insulin delivery, battery life, and CGM data. Provides access to various functionalities like bolus calculator and settings.
2	Setup and charge the pump	Insulin Pump Setup (UC2)	Device, MainWindow, MainWindow.ui	Simulate charging; check battery percentage increase and	Handles charging of the pump, monitoring the battery percent, and simulating battery consumption.

				alert functionality.	
3	Create, read, update, delete personal profiles	Creating a Profile (UC1)	ProfileManager, Profile, Device, MainWindow, Mainwindow.ui	Execute profile management operations; verify data integrity and UI updates.	Manages the creation, retrieval, updating, and deletion of user profiles, linking basal rates, carb ratios, and other critical settings.
4	Manual bolus delivery	Delivering Insulin through Manual Bolus (UC9)	Device, MainWindow, Mainwindow.ui, User	Perform bolus delivery; monitor insulin levels and bolus buffer status.	Controls insulin bolus delivery including start, calculation, and cancellation through manual interaction and monitoring bolus buffer status.
5	Start, stop, and resume insulin delivery	Starting Insulin Delivery (UC4), Stopping Insulin Delivery (UC5), Resuming Insulin Delivery (UC6)	Device, MainWindow, Mainwindow.ui, User	Test starting, stopping, and resuming delivery; check system logs for accuracy.	Manages the basal rate simulation, emergency bolus, and insulin delivery, including starting and stopping these processes.
6	Data storage for pump operation history	Displaying Pump History (UC8)	MainWindow, Device, MainWindow.ui, HistoryManager	Review historical data display for accuracy and completeness in chart form.	Utilizes the charting features to visualize insulin delivery history and glucose level changes over time.
7	Handle malfunctions and alerts	Handling Low Battery Alert (UC10), Handling Occlusion Alert (UC11), Handling CGM Disconnection Alert (UC12), Handling Pump	Device, MainWindow, MainWindow.ui	Trigger alerts manually; observe responses and automatic handling of emergencies.	Monitors and handles low insulin and battery levels, providing warnings and managing emergency boluses as necessary.

		Shutdown (UC13)			
8	Graphing of usage data	Displaying Pump Info (UC7), Displaying Pump History (UC8)	Device, MainWindow, MainWindow.ui	Generate usage data reports; verify graph accuracy and data responsiveness.	Uses charting tools to provide graphical representations of usage metrics, enhancing data analysis and insights into patient glucose control.
9	Implement PIN-based security	Insulin Pump Setup (UC2)	MainWindow, Device, MainWindow.ui, User, Profile	Test PIN setup and access restrictions; attempt unauthorized access simulations.	Manages user interaction for setting up PIN-based security measures, enhancing privacy and security.
10	Simulate pump's hardware interaction	Interacting with Pump Interface (UC3)	Device, MainWindow, MainWindow.ui	Simulate hardware inputs; validate system reactions and update display accordingly.	Simulates physical interactions with the pump's hardware, such as managing battery and insulin warnings, and responding to user interactions.
11	Log and troubleshoot errors	Handling Pump Shutdown (UC13)	MainWindow, Device, MainWindow.ui	Introduce faults; monitor log entries and error handling efficacy.	Utilizes the system's logging capabilities to record activities and identify issues for troubleshooting.
12	Provide real-time updates on battery and insulin	Interacting with Pump Interface	Device, MainWindow, MainWindow.ui	Continuously monitor battery and insulin updates; assess real-time data accuracy.	Monitors and updates the UI with real-time data on battery and insulin levels using Device's status-checking functions.

13	Extend bolus over a period	Delivering Insulin through Manual Bolus (UC9)	Device, MainWindow, MainWindow.ui, User	Conduct extended bolus delivery; verify timing and dosage consistency.	Handles extended and quick bolus management, ensuring delivery over set periods or immediate needs.
14	Optimize treatment based on CGM feedback	Starting (UC4), Stopping (UC5), or Resuming Insulin (UC6)	Device, MainWindow, MainWindow.ui, User	Test adaptive insulin delivery; analyze system response to changing glucose levels.	Adjusts insulin delivery dynamically based on real-time CGM feedback and emergency conditions.
15	Support for touchscreen navigation and input	Interacting with Pump Interface (UC3)	MainWindow, Device, MainWindow.ui	Evaluate touchscreen functionality; perform usability testing with multiple users.	Manages user inputs via touchscreen, enhancing interface navigation and input responsiveness.

-X-X-X-X-X-X-X-