COMP 3004 - Team Project

By: Daniel, Se Won, Ashwin, Hendry, Teja
Updated 2025-04-08

This project is delayed.

As with many efforts in software engineering beyond the most simple of projects, it is impossible for planning to account for everything that might go in a project. The same applies with this project, as many unexpected circumstances arose during the development phase.

Overall, our timeline has been tight to begin with, especially for the part where we develop logic for passing and controlling data between our various screens. We initially thought putting the system-wide data in the main window would be a good solution, but that ended up being very impractical during development. We had to change our diagrams to reflect the change, and comb through every technical detail that we assumed was "simple to do."

And throughout development, that has been happening constantly, with us not realizing the scope of each feature, regarding it as easier than it actually is. In the end, it chewed through our flex week as we met and discussed multiple times on what to do. As of writing, we are still working on the data logic, but making good progress. The new schedule will start from the due date with everything we've done, and outlining what we will do in the next 2 weeks. Team responsibilities remain the same.

Updated Schedule From April 8th Onwards:

April 8th	Currently completed: - Design documentation - Repo setup - Dumb GUI without functionality - About 50% completed with facilitation for data handling and UI
April 8th - 13th	Minimal Success Scenario completed - Reading data from hardware components - Control IQ technology - Full UI completion
April 14th - 16th	Error handling Graphs UI enhancements
April 17th - 18th	Traceability matrix
April 19th - 22nd	Bug testing Flex time
April 22nd	Due date, submit assignment

Team Responsibilities:

Design Documentation:
- **Use Cases**: Teja
- **UML Class Diagram**: Daniel
- **UML Sequence Diagrams**:
    - *Normal*: SeWon
        - *Starting pump and administering insulin*
        - Set personal profile
        - Manual bolus delivery
    - *Safety*: Ashwin
        - Low insulin before starting pump
        - CGM disconnected during pumping
- **UML State Machine Diagrams**: Hendry

Dumb GUI (No functionality)

1. **Home** Screen: Hendry
    a. Also external toggles like USB, power button, bolus button
2. **Bolus**: Se Won & Daniel
3. **Options** -> adding profiles and changing data within the profiles: Teja & Ashwin

Logic For Single Profile: Teja & Ashwin

Reading data from hardware components: Hendry

UI Interactions: ALL

Logging(log screen): Se Won
- Log class with functions to send/retrieve logs. Handles exporting logs to file for historical data

Control IQ technology: Daniel

Graphs: Daniel

Configurable Profiles Complete: Teja
- Give users the option to alter their profile or other profiles on the device.

Error Handling:
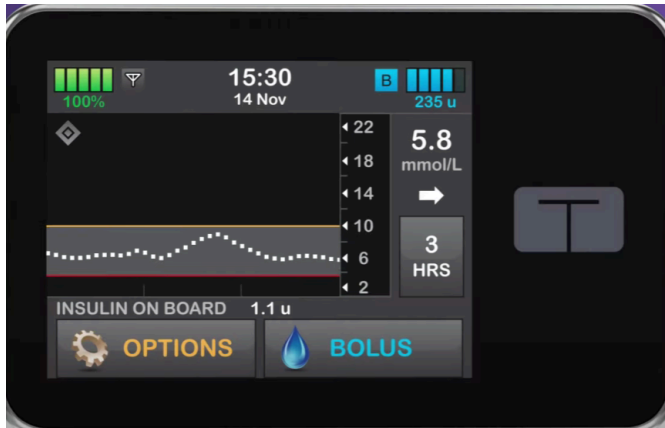Hendry (non-pumping related errors)
Se Won, Teja (errros during pumping)

Traceability Matrix: ALL (while they are working on their own parts)

Requirements (for internal tracking):

Project UI:
- Hardware (device)
- Software (Screen)
- These two are contained within group boxes within the QT GUI editorr
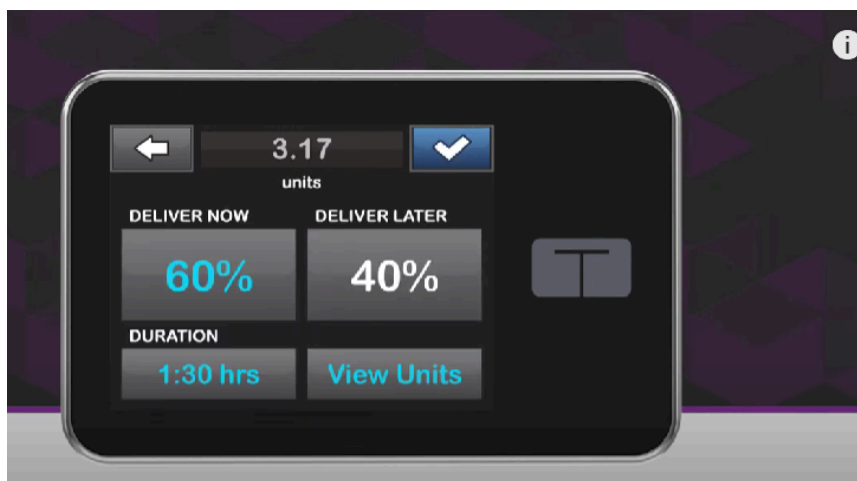
Home Screen

- *Monitor Insulin Delivery* (Upper Right Corner)
    - Remaining Insulin in 300-unit cartridge
    - Insulin active in body after previous bolus injections
- *Monitor Battery Life* (Upper left corner)
- *Continuous Glucose Monitoring* (CGM)
- *Navigation Buttons*
    - Bolus button
        - Directs to bolus calculator
    - Options button
        - Insulin delivery settings
- *Home button*

Options Menu (Insulin Delivery Settings)

- Manage personal profiles (CRUD access)
    a. Name
    b. Basal rate
    c. Carb ratios
    d. Correction factors
    e. Target glucose levels
-

Bolus Calculator:

- Input boxes (Automatically populate input boxes when opening "bolus" from home screen and can be changed manually)
    - Carbohydrate intake
    - Current blood glucose

- Calculator suggests appropriate dose based on programmed settings (insulin sensitivity, target glucose levels)
- User can adjust dosage and delivery time
    - Quick delivery for immediate correction of high glucose
- User can cancel or stop bolus delivery if needed

LOGGING:
The system logs information such as
1. basal rates,
2. bolus injections
3. insulin duration
4. correction factors
5. *Common issues such as low battery, low insulin, or CGM disconnection trigger alerts*

Classes

1. Profile (class)
2. MainWindow (class)
    a. QStackedWidget
        i. BolusUI
        ii. HomeUI
        iii. OptionsUI
    b. Profiles (vector containing Profile)
3. ControlIQ (static class)
4. Logger (static class)
    a. getLogs (function)
    b. addLog (function)
    c. saveLog (function)
5. Bolus (static class)