Traceability Matrix

Note: “References requirement #n” represents the corresponding section in the Team Specification file.
For example, “References requirement #2” refers to the section “Getting Started” in Team Specification file.

ID	Requirement	Use Case(s)	Design Element(s)	Test By	Description
1	Users can create a new profile by navigating to the personal profiles section and entering critical settings like basal rates, carbohydrate ratios, correction factors, and target glucose levels. Each profile can be named based on specific routines, such as “Morning Routine” or “Exercise Mode.” Once created, profiles can be reviewed and updated as needed. If a profile is no longer needed, it can be deleted to streamline options.(References requirement #3)	Create a personal profile(use case #4), Edit personal profile(use case #5), View Personal Profile(use case #6), Delete a Personal Profile(use case #7)	settingsscreen.ui, mypumpscreen.ui	Run the simulation in QT and try and create a new personal profile. Then, edit the newly created profile, save the changes and view the profile to verify those changes. The same process would be done to delete a profile.	Using Qt’s built-in UI, users can create or edit profiles by navigating to the personal profiles section via the “Options” button on the home screen. There, they can enter or update key settings like carb ratio, correction factor, and target glucose, and save changes by clicking the “Save Profile” button. To delete a profile the user can select a profile and press the “delete” button to remove a profile. When they press the button, a pop up will appear to verify that the user wants to delete a given profile.
2	The t:slim X2 insulin pump maintains detail ed records of insulin delivery events, which users can review to track their treatment progress. The system logs information such as basal rates, bolus injections, insulin duration, and correction factors. By accessing the current status screen, users can	View Insulin Delivery History(Use case #12)	loggerscreen.ui, logger.cpp	When performing tests on any of the other items in this traceability matrix, go to “history” in the “options” screen. In the logging screen, users can view different logs that were made by the insulin pump to indicate different	The Logger and LoggerScreen classes work together to track activity on the t:slim X2 insulin pump. Logger handles core logging through static methods like log() and info(), capturing events such as insulin deliveries and pump alerts. LoggerScreen, built with Qt, displays this data using a QTextEdit widget. The vertical layout makes it easy to

ID	Requirement	Use Case(s)	Design Element(s)	Test By	Description
	view recent events like the time and amount of the last bolus, changes in basal rates, or alerts triggered by CGM readings. This data is crucial for identifying patterns or irregularities in glucose control and can assist healthcare providers in optimizing treatment plans. The stored history allows users to trace back specific events to understand how insulin was administered during different situations. (Reference Requirement 6)			events and alerts.	review treatment history and spot insulin trends by keeping track of different events and warnings produced by the insulin pump. This could provide information healthcare providers need in order to obtain better insights on how a patient can manage their glucose.
3	Bolus delivery affects stored insulin level (Reference Requirement 4)	UC #8: Deliver a Manual Bolus	Data, BolusScreen (confirm insulin delivery), HomeScreen (insulin bar)	When the user authorizes a bolus, the insulin bar should deplete according to the amount specified by the bolus screen.	When a timed/instant bolus is delivered, BolusScreen uses the values entered by the user (carbs and glucose level if they wish) and adjusts the insulin level in data over time
4	Bolus delivery can be extended over time (Reference Requirement 4)	UC #8: Deliver a Manual Bolus	Data, BolusScreen, HomeScreen	Observe bolus decreasing over time in the home screen in the stored insulin bar.	Before a user authorizes a bolus, they can choose to extend the bolus over time and choose the percentage and duration in simulated hours (1h = 12 real seconds) BolusScreen will modify the storedInsulinLevel parameter within

ID	Requirement	Use Case(s)	Design Element(s)	Test By	Description
					Data every second to deliver a bolus over time. This timed delivery can be viewed within the home menu.
5	Bolus delivery affects blood glucose (Reference Requirement 4)	UC #8: Deliver a Manual Bolus	Data, BolusScreen, HomeScreen, Data	When the user authorizes a bolus, the insulin bar should deplete according to the amount specified by the currently loaded profile's targetGlucose parameter.	When a timed/instant bolus is delivered, BolusScreen uses the targetGlucose parameter within the currently loaded profile to match the user's current blood glucose to it.
6	Determining how much Bolus to provide is determined through real-time data reading by Control IQ technology. (References requirement 4)	UC #9: Start Insulin Delivery, UC #11: Resume Insulin Delivery, UC #8: Deliver a Manual Bolus	BolusScreen, HomeScreen, Data	When we manually change the CGM value in the insulin simulation, we check if the blood glucose level is changed in response. This would indicate that control IQ is dynamically adjusting the insulin delivery rate.	Control IQ's real-time bolus system works by predicting where your blood sugar is heading and adjusting insulin automatically. It starts with a method called predictGlucose(), which looks at recent CGM data to estimate your glucose levels 30 minutes ahead. Then, handlePredictedGlucose() uses those predictions to change how much insulin is given. If your blood sugar is low, it can stop insulin. If it's high, it can increase the insulin delivery rate. The system uses different levels of adjustment depending on how high or low your glucose is. It also uses your personal settings and real-time data to make these adjustments.
7	Pump displays Battery indicator (level, recharge status). (Reference Requirement 1)	UC 1: View Home Screen	Data, HomeScreen (batteryBar,	Observe Battery display on Home Screen. Use	Data stores battery level and charging status. HomeScreen displays the status using a progress

ID	Requirement	Use Case(s)	Design Element(s)	Test By	Description
			batteryText), MainWindow (battery timer, charging button)	input/charger button to change status.	bar and text label, updated via Data signals. MainWindow includes controls for simulation and timers for discharge/charge.
8	Pump displays Insulin fill gauge (300-unit level). Reference Requirement 1)	UC 1: View Home Screen	Data, HomeScreen (insulinBar, insulinText)	Observe Stored Insulin display on Home Screen. Use input to change level.	Data stores stored insulin level. HomeScreen displays the level using a progress bar and text label, updated via Data signals.
9	Pump displays Insulin on Board (IOB) value. Reference Requirement 1)	UC 1: View Home Screen, UC 8: Deliver a Manual Bolus	Data, HomeScreen (insulinOnBoardLabel)	Observe IOB display on Home Screen.	Data stores IOB value (placeholder for now). HomeScreen displays the value using a label, updated via Data signals.
10	Pump displays current CGM Glucose Value (numeric). Reference Requirement 1)	UC 1: View Home Screen, UC 8: Deliver a Manual Bolus	Data, HomeScreen (CGMValueLabel)	Observe CGM value label on Home Screen.	Data stores current CGM value and connection status. HomeScreen displays the value using a label, updated via Data signals.
11	Pump displays CGM trend graph. Reference Requirement 1)	UC 1: View Home Screen, UC 8: Deliver a Manual Bolus, UC 14: Graph Insulin Delivery and CGM Data	Data, HomeScreen (graph), GraphWidget	Observe graph display on Home Screen.	Data stores historical CGM readings. GraphWidget reads this data and draws the graph. HomeScreen contains the GraphWidget.
12	CGM graph points/lines are colored based on connection status when reading was taken. (Reference Requirement 1)	UC 14: Graph Insulin Delivery and CGM Data	Data (CgmReading struct), GraphWidget (paintEvent)	Connect/Disconnect simulated CGM and observe graph color changes.	Data stores connection status with each reading. GraphWidget uses this status to color individual points/segments during painting.
13	CGM graph time span can be toggled (1h, 3h, 6h).	UC 14: Graph Insulin Delivery and CGM Data	Data (graph hours), HomeScreen	Click Hour button on Home Screen and observe graph scaling.	HomeScreen button changes graph hours in Data. GraphWidget reads graph hours from Data to scale the

ID	Requirement	Use Case(s)	Design Element(s)	Test By	Description
	(Reference Requirement 1)		(hourButton), GraphWidget (paintEvent)		X-axis display.
14	Pump displays simulated date and time. (Reference Requirement 1)	UC 1: View Home Screen, Project Spec (real-time updates)	Data (simulatedDateTime), HomeScreen (dateTimeLabel, dateTimeTimer)	Observe date/time display on Home Screen.	Data stores and advances simulated time. HomeScreen displays this time, updated by a timer and Data signals.
15	Holding power button from off initiates power on sequence. (Reference Requirement 2)	UC 2: Power on the Pump, Project Spec (startup sequence)	MainWindow (powerButton, powerHoldTimer, slots), Data (currentScreen)	Hold Power button >1s from Truly Off state. Observe transition.	MainWindow detects long press using a timer (1s in your code). It sets the Data state to trigger the boot sequence.
16	Short pressing power button toggles between On and Standby states. (Reference Requirement 2)	UC 3: Power off or Sleep mode	MainWindow (powerButton, powerHoldTimer, slots), Data (currentScreen)	Short press Power button from On state (-> Standby), then short press from Standby state (-> Wake/PIN).	MainWindow detects short press using a timer. It sets the Data state to toggle between functional screens and OffScreen (Standby state).
17	Power On sequence displays startup animation/progress bar. (Reference Requirement 2)	UC 2: Power on the Pump, Project Spec (startup sequence)	Data (currentScreen), OffScreen (offScreenStack, bootingPage, bootProgressBar, bootTimer)	Long press Power button from Truly Off. Observe boot animation.	When Data state is Booting, MainWindow shows OffScreen. OffScreen internally switches to the Booting page and manages the progress bar animation.
18	Pump requires PIN entry after power on or waking from Standby. (Reference Requirement 2)	UC 2: Power on the Pump, Project Spec (PIN lock)	Data (currentScreen, pinCode), OffScreen (offScreenStack, pinLockPage, PIN	After boot or wake, observe transition to PIN screen. Enter correct/incorrect PIN.	Data state transitions trigger PIN screen. OffScreen shows PIN page. Entering correct PIN (verified against Data) allows setting Data state to last active screen.

ID	Requirement	Use Case(s)	Design Element(s)	Test By	Description
			input/buttons)		
19	PIN can be entered via on-screen keypad or keyboard. (Reference Requirement 2)	UC 2: Power on the Pump	OffScreen (pinInputLineEdit, keypad buttons, slots)	Use on-screen keypad or physical keyboard to enter PIN on PIN screen.	OffScreen connects keypad button/QLineEdit signals to slots that populate the PIN input field.
20	Pump displays Low Battery alert. (Reference Requirement 7)	UC 13: Handle Pump Errors and Alerts, Project Spec (low battery alert)	Data (batteryLevel, signal), MainWindow (slot), PopupManager, PopupWidget	Set battery level < 20 via input. Observe popup alert.	Data signal for low battery triggers a slot in MainWindow, which uses PopupManager to display a modal popup.
21	Home Screen includes navigation buttons for Bolus and Options. Reference Requirement 1)	UC 1: View Home Screen, UC 8: Deliver a Manual Bolus, Project Spec	HomeScreen (bolusButton, optionsButton), Data (currentScreen)	Click Bolus/Options buttons on Home Screen.	HomeScreen buttons set Data state to trigger screen navigation.
22	Insulin delivery can be stopped/resumed (Reference Requirement 5)	Use Case #10: Stop Insulin Delivery Use Case #11: Resume Insulin Delivery	SettingsScreen, HomeScreen	Observe if insulin delivery on the home screen stops when hitting the stop insulin delivery button within the settings menu	SettingsScreen changes a boolean in Data which prevents the delivery of insulin within Data where the stored insulin value is held.