

Traceability Matrix

ID	Requirement	Related Use Case	Fulfilled By	Implemented By	Tested By
1	System loads user profiles and logs, initializes battery, CGM reader, insulin reservoir, pump controller, and displays the login UI	UC1: Initialize Pump System	Device PumpController CGMReader BatteryManager InsulinReserve Profile Datalogger UserInterface Login	device.cpp/.h/.ui pumpcontroller.cpp/.h cgmreader.cpp/.h batterymanager.cpp/.h profile.cpp/.h datalogger.cpp/.h userinterface.cpp/.h/.ui login.cpp/.h/.ui	Manually power on. Verify home screen appears, verify profiles.json & logs.json are properly created. Verify that history has loaded and displays logs correctly, also verify in settings that the default profile correctly initialized.
2	The system allows users to create, update, delete, and select personal profiles	UC2: Create or Modify Personal Profile	Profile Settings	profile.cpp/.h settings.cpp/.h/.ui	Create multiple profiles with varying basal/carb/correction/target values. Verify profile creation was successful. Select profiles and observe changes in behaviour. Change values and hit update; verify changes were applied successfully. Delete a profile and verify that it is removed from the UI and profiles.json.
3	The system lets the user deliver manual bolus doses, based on the correct formula with optional manual override	UC3: Deliver Manual Bolus	BolusCalculator PumpController	boluscalculator.cpp/.h/.ui pumpcontroller.cpp/.h	In BolusCalculator, enter glucose and carb values, click Calculate and confirm the dose matches the formula. Click Deliver and accept the confirmation; verify Home's bolus status updates, also verify actual bolus delivery. Also test with manual override and extended bolus delivery options.

4	The system calculates a recommended bolus dose given glucose and carb inputs	UC4: Calculate Bolus Dose	BolusCalculator	boluscalculator.cpp/.h/.ui	Verify that calculated bolus dose matches the correct formula in multiple tests.
5	The system monitors CGM readings and automatically adjusts basal insulin delivery	UC5: Monitor and Adjust Insulin Delivery	ControlIQAlgorithm CGMReader IOBTracker Bloodstream PumpController	controliqalgorithm.cpp/.h cgmreader.cpp/.h iobtracker.cpp/.h bloodstream.cpp/.h pumpcontroller.cpp/.h	Observe simulation and verify that insulin delivery changes based on glucose changes that match expected behaviour. Modify profile values and verify changes in the delivery match expectations.
6	The system suspends active bolus delivery under unsafe conditions, and resumes once safe	UC6: Suspend and Resume Insulin Delivery	PumpController ControlIQAlgorithm	pumpcontroller.cpp/.h controliqalgorithm.cpp/.h	Manually deliver large dose of insulin or modify profile to simulate glucose < 3.9. Verify that the pump is disabled, and the UI reflects the change. Also manually test the cancelling of an extended dose and verify success.
7	The system detects critical conditions and raises alerts and logs warnings	UC7: Handling Pump Malfunction or Alert	BatteryManager InsulinReserve CGMReader UserInterface PumpController Alert Datalogger	batterymanager.cpp/.h insulinreserve.cpp/.h cgmreader.cpp/.h pumpcontroller.cpp/.h alert.cpp/.h/.ui datalogger.cpp/.h	Wait until battery level drops below 15%, verify that alert appears and the log shows the warning. Simulate CGM disconnect with the disconnect button, verify alert and log. Modify profile and observe high glucose alert and log entry.
8	The system logs all events, glucose & insulin readings to persistent storage	UC8: View History and Data Logs	Datalogger History	datalogger.cpp/.h history.cpp/.h/.ui	Trigger an alert or manual bolus and observe log entries in history and logs.json. Test eventType filters and search functionality in history.

9	The system requires entry of a correct PIN at startup to prevent unauthorized access	UC9: Unlocking Pump with PIN	UserInterface Login	userinterface.cpp/.h/.ui login.cpp/.h/.ui	Enter any pin other than “1234” on the pin screen and observe the “Incorrect Pin” alert. Enter “1234” and observe that the UI transitions to the Home screen.
10	The system enables seamless navigation between screens	UC10: Navigating Between Screens	UserInterface Home BolusCalculator History Login Settings	userinterface.cpp/.h/.ui home.cpp/.h/.ui boluscalculator.cpp/.h/.ui history.cpp/.h/.ui login.cpp/.h/.ui settings.cpp/.h/.ui	From the Home screen navigate to all menus using the provided GUI buttons. Ensure that the (T) logo always successfully returns to the Home screen.