ID	Requirement	Related Use Case(s)	Implemented By	Tested By	Test Description
REQ-01	The system shall allow users to create new profiles with custom settings	Use Case 1: Create a New Profile	- Simulation::createProfile() - InsulinPump constructor	- ProfileCreationTest - IntegrationTest_ProfileManagement	- Verify profile is created with specified name - Verify default settings are applied - Test duplicate name handling
REQ-02	The system shall allow users to delete existing profiles	Use Case 2: Delete a Profile	- Simulation::deleteProfile()	- ProfileDeletionTest - IntegrationTest_ProfileManagement	- Verify profile is removed from system - Verify confirmation before deletion - Test handling when current profile is deleted
REQ-03	The system shall allow users to switch between profiles	Use Case 1: Create a New Profile	- Simulation::switchProfile()	- ProfileSwitchingTest - IntegrationTest_ProfileManagement	- Verify active profile changes - Verify settings update to reflect new profile - Test invalid profile selection
REQ-04	The system shall calculate and deliver manual bolus insulin based on glucose level and carbohydrate intake	Use Case 3: Deliver a Manual Bolus	- InsulinPump::manualBolus() - InsulinPump::deliverBolus()	- ManualBolusTest - IntegrationTest_InsulinDelivery	- Verify bolus calculation based on inputs - Verify insulin reservoir is updated - Test delivery with insufficient insulin
REQ-05	The system shall deliver basal insulin automatically at configured rates	Use Case 4: Automatically Deliver Basal Insulin	- InsulinPump::setBasalRates() - Simulation::updateSimulation()	- BasalDeliveryTest - IntegrationTest_InsulinDelivery	- Verify basal insulin delivery at scheduled intervals - Verify correct amounts delivered - Test basal rate changes

REQ-06	The system shall maintain a historical record of all insulin delivery events	Use Case 5: View Insulin Pump History	- InsulinPump::logBolusEvent() - InsulinPump::getHistory() - InsulinPump::printRecords()	- HistoryLoggingTest - IntegrationTest_DataStorage	- Verify events are logged accurately - Verify all required data is captured - Test history retrieval and formatting
REQ-07	The system shall allow users to view the insulin pump history	Use Case 5: View Insulin Pump History	- Simulation::storeCurrentProfileRecords() - Simulation::printAllLogs()	- HistoryViewingTest - IntegrationTest_DataStorage	- Verify history display functionality - Verify filtering and sorting options - Test with large history datasets
REQ-08	The system shall allow users to suspend and resume insulin delivery	Use Case 6: Suspend/Resume Insulin Delivery	- InsulinPump::suspendInsulinDelivery() - InsulinPump::resumeInsulinDelivery()	- SuspendResumeTest - IntegrationTest_SafetyFeatures	- Verify insulin delivery stops when suspended - Verify delivery resumes properly - Test bolus attempts during suspension
REQ-09	The system shall provide PIN security for critical settings	Use Case 3: Deliver a Manual Bolus Use Case 6: Suspend/Resume Insulin Delivery	- InsulinPump::password() - InsulinPump::setPIN()	- PINSecurityTest - IntegrationTest_SafetyFeatures	- Verify PIN validation for sensitive operations - Test incorrect PIN handling - Verify PIN change functionality
REQ-10	The system shall monitor and report battery status	Use Case 7: Battery Status Check	- InsulinPump::checkBatteryStatus()	- BatteryMonitoringTest - IntegrationTest_SafetyFeatures	- Verify low battery alerts - Test critical battery threshold actions - Verify battery level display accuracy

REQ-11	The system shall check insulin levels and alert when low	Use Case 3: Deliver a Manual Bolus Use Case 4: Automatically Deliver Basal Insulin	- InsulinPump::checkInsulinLevel() - InsulinPump delivery methods	- InsulinLevelTest - IntegrationTest_SafetyFeatures	- Verify low insulin alerts - Test delivery attempts with low insulin - Verify proper threshold detection
REQ-12	The system shall automatically adjust insulin delivery based on CGM data (Control-IQ)	Use Case 4: Automatically Deliver Basal Insulin	- InsulinPump::integrateCGM() - Control-IQ algorithm	- ControlIQTest - IntegrationTest_AutomaticAdjustment	- Verify automatic suspension for low glucose - Test increased delivery for high glucose - Verify proper response to glucose trends
REQ-13	The system shall detect and handle pump malfunctions and errors	All Use Cases	- Error handling in all components - InsulinPump::logError()	- ErrorHandlingTest - IntegrationTest_SafetyFeatures	- Verify error detection for various scenarios - Test appropriate alert generation - Verify system recovery after errors
REQ-14	The system shall provide data visualization for glucose and insulin trends	Use Case 5: View Insulin Pump History	- Data visualization components - Reporting functionality	- VisualizationTest - IntegrationTest_Reporting	- Verify graph generation for time-series data - Test visualization of different metrics - Verify interactive data exploration
REQ-15	The system shall support extended bolus delivery over time	Use Case 3: Deliver a Manual Bolus	- InsulinPump::deliverExtendedBolus()	- ExtendedBolusTest - IntegrationTest_InsulinDelivery	- Verify immediate portion delivery - Test scheduled portion delivery - Verify proper scheduling and timing