Tested By Column is shown via videos in repoistory

| ID | Requirement | Use Case | Fulfilled By | Description | Implemented By | Tested By |
|---|---|---|---|---|---|---|
| 1 | The pump's home screen is the central hub where users can monitor insulin delivery, battery life, and CGM (Continuous Glucose Monitoring) data. | Model 5 Step 4 | glucoseGraph | Glucose graph shows current glucose in the body overtime | glucose_graph_widget | |
| 2 | Located in the upper-left corner is the battery indicator, which provides real-time updates on the status of the pump's rechargeable battery. | Model 5 Step 3 | Battery (QProgressBar), BatteryPercentLabel (QLabel) | On the menu page, labels and progress bars for the battery are displayed | mainwindow { setupBattery() } | |
| 3 | Opposite to this, the insulin fill gauge in the upper-right corner displays the remaining insulin in the 300-unit cartridge. | Model 5 Step 4 | insulinRemaining | When insulin remaining becomes low enough, a message is displayed. The user can then change the cartridge via the options page. | mainwindow { ChangeCartridgeButton } | |
| 4 | The "Insulin on Board" (IOB) indicator shows how much insulin is still active in the body from previous bolus injections. | Model 5 Step 7 | iobTimer, IOBLabel (QLabel) | Every 60s insulin on board is calculated and the label is updated | mainwindow, bolusScreen { ConfirmButton } | |
| 5 | The home screen includes key navigation buttons such as the bolus button, which directs users to the bolus calculator. | Model 5 Step 4 | BolusOption (QPushButton) | User presses Bolus button which connects to a function which sets the current widget to BolusScreen | connect(ui->BolusOption … setCurrentWidget (ui->BolusScreen) | |
| 6 | The home screen includes the options button, which provides access to insulin delivery settings, alerts, and system configurations. | Model 5 Step 4 | OptionsButton (QPushButton) | User presses Options button which connects to a function which sets the current widget to OptionScreen | connect(ui->OptionsButton … setCurrentWidget (ui->OptionScreen) | |
| 7 | Users can quickly return to the home screen by tapping the Tandem logo from any screen within the pump. | Model 5 Step 5a | TandemLogo (QPushButton) | User presses Tandem logo which connects to a function which sets the current widget to HomeScreen | connect(ui->TandemLogo … setCurrentWidget (ui->HomeScreen) | |
| 8 | Turning the pump on involves holding the power button until the startup sequence is complete. | Model 5 Step 1 | PowerButton (QPushButton) | User presses power button and the screen turns on, press it again and the screen turns off | connect(ui->PowerButton … ui->BlackBox->*setVisible*( .. ) | |
| 9 | For security, a PIN-based lock screen can be set up to prevent accidental inputs. | Model 5 Step 2 | Lock, pinEdit | When the pump is run the first page is a lock screen which requires the user to enter a 4-digit pin correctly. Default is '1234' | connect(ui->pinEdit, &QLineEdit::textChanged … ) | |
| 10 | The t:slim X2 insulin pump allows users to customize and manage personal profiles that cater to different daily needs, such as varying basal rates during sleep or exercise. | Model 1 | PersonalProfilesButton, PersonalProfilesScreen | Personal Profiles Screen allows users to create multiple customizable profiles to cater to different requirements. | OptionScreen { PersonalProfilesButton } | |
| 11 | Users can create a new | Model 1 | CreateProfileButton, | In the personal profiles screen you | ProfileOptionScreen { | |

| | | | | | | |
|---|---|---|---|---|---|---|
| | profile by navigating to the personal profiles section and entering critical settings like basal rates, carbohydrate ratios, correction factors, and target glucose levels. | Step 7 | ProfileOptionScreen | can click the + which creates a new profile where you can enter critical settings | CreateProfileButton } | |
| 12 | Users might modify a profile to increase basal insulin delivery during high-activity periods or adjust carb ratios for a large meal. | Model 2 Step 4, 6 | EditProfileButton, ProfileOptionScreen | In the personal profiles screen you can click the edit button which allows you to edit the currently selected profile and modify critical settings | ProfileOptionScreen { EditProfileButton } | |
| 13 | If a profile is no longer needed, it can be deleted to streamline options. | Model 2 Step 4a | DeleteProfileButton (QPushButton), ProfileOptionScreen (QPushButton) | In the personal profiles screen you can click the - which deletes the currently selected profile | ProfileOptionScreen { DeleteProfileButton } | |
| 14 | Users initiate a bolus by accessing the bolus calculator via the home screen or bolus button. | Model 6 Step 1 | BolusOption (QPushButton) | User presses Bolus button which connects to a function which sets the current widget to BolusScreen | connect(ui->BolusOption … setCurrentWidget (ui->BolusScreen) | |
| 15 | They can enter their current blood glucose level and carbohydrate intake manually or have these values pulled automatically from the CGM. | Model 6 Step 2 | ConfirmButton (QPushButton) CarbsSpinBox (QSpinBox) GlucoseSpinBox (QSpinBox) | User manually enters carbohydrates intake and can optionally enter glucose level, if not entered will automatically pull glucose level. Clicking confirm will begin the bolus calculation. | connect(ui->ConfirmButton … this)) | |
| 16 | The bolus calculator then suggests an appropriate dose based on programmed settings such as insulin sensitivity and target glucose levels. | Model 6 Step 3 | ConfirmButton (QPushButton) | Uses insulin sensitivity (correction factor) and target blood glucose to calculate appropriate dose administer | mainwindow { calculateSuggestedBolus(bg, carbs) } | |
| 17 | The pump also supports extended boluses, where insulin delivery is spread over a longer period. | Model 6 Step 5a | BolusSettingEditButton (QPushButton) | User enters duration and quantity of bolus | mainwindow { BolusSettingDurationSpinBox } | |
| 18 | The pump supports quick boluses for immediate correction of high glucose. | Model 6 Step 5b | BolusOption | User can click bolus option button from home page to enter quick bolus | mainwindow { BolusScreen } | |
| 19 | The simulation should allow for generating, reading, and graphing of data. | | glucoseGraph | The glucose graph updates live glucose levels for the end user | glucose_graph_widget | |
| 20 | Managing basal insulin delivery is an essential aspect of the pump's functionality, and Control IQ technology simplifies this by dynamically adjusting delivery based on CGM feedback. | Model 3 Step 4 | controlIQ | ControlIQ dynamically increases, decreases basal delivery when too high or low, ceases delivery below 3.9mmol/L and administers a bolus above 1.0mmol/L | control_iq_manager { handleCGM } | |
| 21 | To start insulin delivery, users must select a basal rate from their active personal profile or manually configure it through the options menu. | Model 3 Step 2 | ConfigureBasalButton, ProfileBasalButton | User selects either the configured basal button and enters the basal rate to deliver, or profile basal button which automatically sets basal rate to the rate specified in the active profile. | pumpSettingsScreen { connect(ui->ConfigureBasalButton … ), connect(ui->ProfileBasalButton … ) | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 22 | Once set, the pump continuously delivers insulin at the specified rate unless interrupted. | Model 3 Step 5 | cgmSim, controlIQ | cgmSim creates a new glucose reading every 5s, calling a function that administers insulin | mainwindow { newGlucoseReading, handleCGM } | |
| 23 | Stopping insulin delivery automatically when Control IQ technology detects low glucose levels (below 3.9 mmol/L). | Model 3 Step 4b | controlIQ | controlIQ automatically detects glucose levels below 3.9, suspending insulin delivery | control_iq_manager { handleCGM } | |
| 24 | Stopping insulin delivery manually | Model 3 Step 4a | Pump Settings Page, StopBasalButton, controlIQ | User suspends insulin via StopBasalButton on the pump settings page, suspending insulin delivery | mainwindow { StopBasalButton, setBasal(0) } | |
| 26 | Resuming insulin delivery involves restoring the previous basal rate or switching to an updated profile when glucose levels stabilize. | Model 3 Step 4c | controlIQ | controlIQ automatically detects when glucose levels have surpassed target blood glucose, resuming insulin delivery | control_iq_manager { handleCGM } | |
| 27 | The t:slim X2 insulin pump maintains detailed records of insulin delivery events, which users can review to track their treatment progress. | Model 3 Step 4b4 | history | On bolus and basal delivery, active profile and current CGM data is stored into the history database | history_screen | |
| 29 | By accessing the current status screen, users can view recent events like the time and amount of the last bolus, changes in basal rates, or alerts triggered by CGM readings. | | StatusButton, StatusScreen | When the user selects the status button it directs the user to the status screen which displays all status information | mainindow { connect(ui->StatusButton … setCurrentWidget (ui->StatusScreen) | |
| 30 | The stored history allows users to trace back specific events to understand how insulin was administered during different situations. | Model 4 | history | The history module automatically adds data for every event that occurs, allowing users to go back and see past events | optionsScreen { HistoryButton } | |
| 31 | A low battery warning might prompt users to recharge the pump. | Model 5 Step 1b | batteryLevel | When the battery is below or equal to 20% a message is displayed instructing the user to charge the pump. | mainwindow { updateBattery() } | |