



SOFTWARE SPECIFICATIONS

Painkiller Injection System

Table of Contents

System Architecture..... 2

Software Specifications..... 3

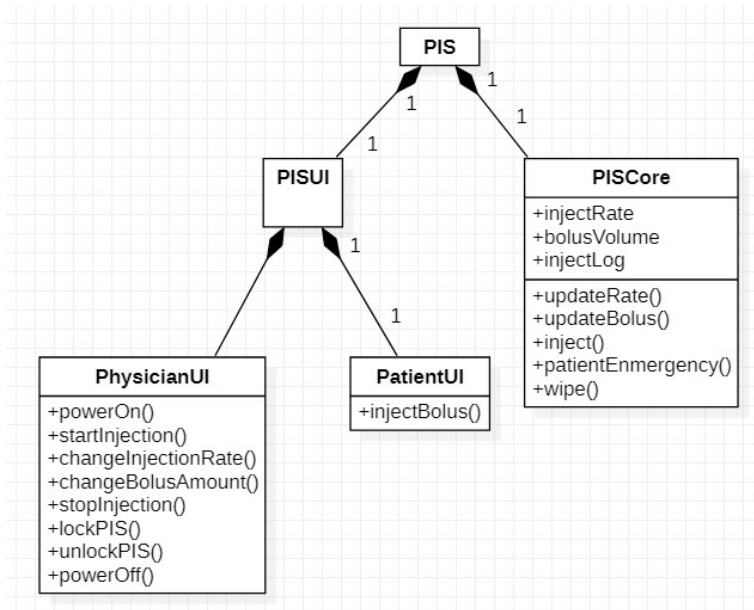
 S1: PISUI implementation 3

 S2: PISCore implementation 5

 S3: PatientUI implementation 6

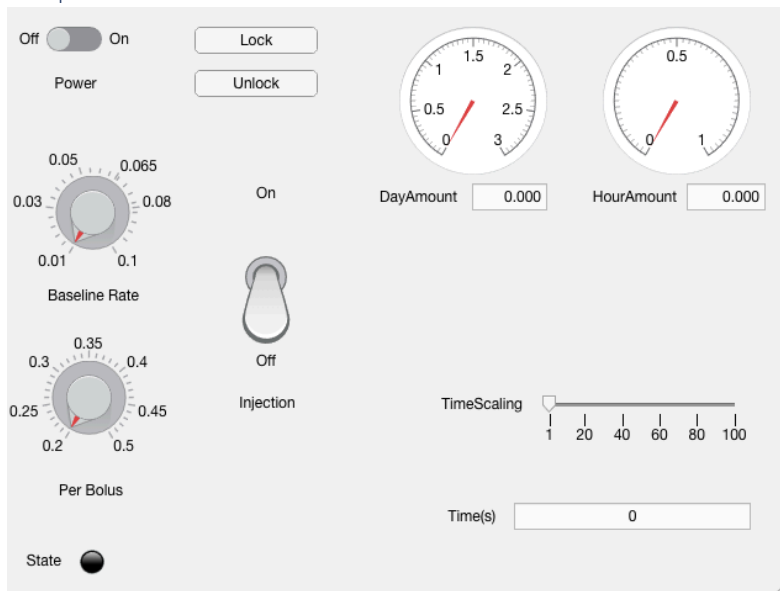
System Architecture

The system architecture is shown below:



Software Specifications

S1: PISUI implementation



S1.1: Power

- S1.1.1: PowerOn
 1. PowerOn Switched
 2. Enable all the availability of buttons
 3. Instantiate the PatientUI
 4. Initialize and begin the global timer t
- S1.1.2: PowerOff
 1. If Not locked and PowerOff Switched
 2. Disable all the availability of buttons
 3. Wipe all the log generated during Powering ON
 - a. Stop global timer t and delete it
 - b. Set HourAmount, DayAmount, TimePassed to 0
 - c. Set baselinerate, PerBolusKnob to Default Minimized Value
 - d. Stop Injection
 - e. Wipe all the injection_log information
 - f. Set TimeScaling to 1
 - g. Close PatientUI

批注 [Z11]: Expand major use cases

S1.2: Safety Lock

- S1.2.1: Lock
 1. If PIS is Powering ON and Not Locked and Lock Button Pushed
 2. Disable all the availability of Buttons except PatientButton and UnlockButton
- S1.2.2: Unlock
 1. If PIS is Powering ON and Locked and Unlock Button Pushed
 2. Enable all the availability of Buttons. Disable LockButton.

S1.3: Injection

- S1.3.1: start Inject and stop inject
 1. Injection is Switched ON
 2. Execute the Following Code Every 1 second
 - a. Get Up-to-date Injection Rate
 - b. Check if this adding will lead to OverDrug, if not:
 - a) Convert the Rate unit to (ml/second) and add it to inject_log, which is a numeric list
 - b) Update DayAmount and HourAmount Value on UI

Else:

 - a) Let the Powerlight be Yellow
 - 3. Stop inject
- S1.3.2: Change Baseline Rate
 1. Baseline Rate is Changed
 2. Update the Baseline Rate UI
- S1.3.3 Change Bolus Amount
 1. Bolus Rate is Changed
 2. Update the Bolus Rate UI
- S1.3.4 Inject Bolus
 1. Power is ON
 2. Check if this adding will lead to OverDrug, if not:
 - a) Add this amount to inject_log
 - b) Update DayAmount and HourAmount

S1.4: Status Indicator

- S1.4.1 Status of Indicator
 1. Black: PIS is Powered Off
 2. Green: PIS is Powered On and has not reached its Hour/Day Limitation
 3. Yellow: PIS is Powered On and reached its Hour/Day Limitation

S2: PISCore implementation

S2.1 Initiate PISCore

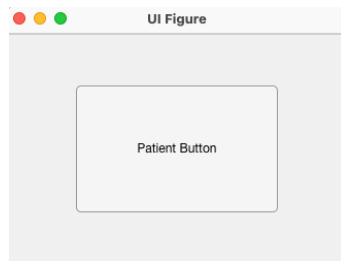
Property:

- a) Injection Rate initiate to 0.01
- b) Bolus_Volume initiate to 0.2
- c) Inject_log initiate to an empty list []

S2.2 Safety Check

- S2.2.1 Check set rate
 1. Set the availability of Injection Knob to [0.01 ,0.1]
 2. No invalid input is allowed.
- S2.2.2 Check injection State
 1. Every time before an injection is executed, call on safety Check
 2. Safety Check
 - a) Check if this injection will lead HourAmount > 1, if yes, tag OverDrug
 - b) Check if this injection will lead DayAmount > 3, if yes, tag OverDrug
 3. An OverDrug Tag will refuse any injection to happen, even if InjectionSwitch is ON.
 4. An OverDrug Tag will set the Status LED to Yellow
- S2.2.3 Check Bolus Legality
 1. Every time Patient Push Patient Button, call on Safety Check
 2. Safety Check
 - a. Check if this bolus will lead HourAmount >1, if yes, Refuse Bolus
 - b. Check if this bolus will lead DayAmount >3, if yes, Refuse Bolus
- S2.2.4 Injection Log
 1. Injection log contains injection history of every second
 2. Every Second, log the actually happened injection amount to it
- S2.2.5 HourAmount Statistics
 1. Every Time Injected, Add Injection Amount to HourAmountValue
 2. Every Second, Minus the Injection Amount happened 3600 seconds ago (if any)
 3. Update HourAmount UI
- S2.2.6 DayAmount Statistics
 1. Every Time Injected, Add injection Amount to DayAmountValue
 2. Every Second, Minus the Injection Amount happened 86400 seconds ago (if any)
 3. Update DayAmount UI

S3: PatientUI implementation



S3.1 Patient Button

- S3.1.1
 1. Patient Button Pushed
 2. Call on SafetyCheck, if safe
 - a. Get up-to-date Bolus Amount
 - b. Add Amount to Injection_log
 - c. Update HourAmount, DayAmount UI