Group 7

Author: Wenlin Zhu

Software Requirements

Painkiller Injection System

Table of Contents

[System Objective 2](#_Toc10411767)

[Domain Analysis 2](#_Toc10411768)

[System Architecture 4](#_Toc10411769)

[Use Cases 5](#_Toc10411770)

[Software Requirements 5](#_Toc10411771)

[R1: PhysicianUI 6](#_Toc10411772)

[R2: PatientUI 6](#_Toc10411773)

[R3: InjectProcessor 6](#_Toc10411774)

[R4: InjectDB 6](#_Toc10411775)

## System Objective

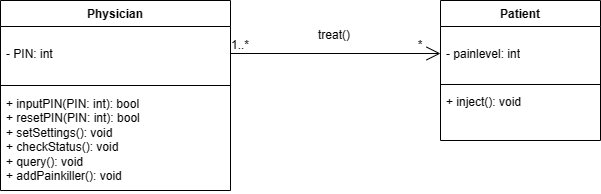
Introducing our Painkiller Injector--a revolutionary device designed to alleviate pain effectively and efficiently. With customizable dosage limits set by the attending physician, it automatically calculates and administers the appropriate painkillers to patients. Should the patient experience intense pain, they have the option to press a button for a bolus.

Featuring both a physician interface for setting limits and monitoring, as well as a patient interface for ease of use, our Painkiller Injector ensures precise and tailored pain management for every individual.

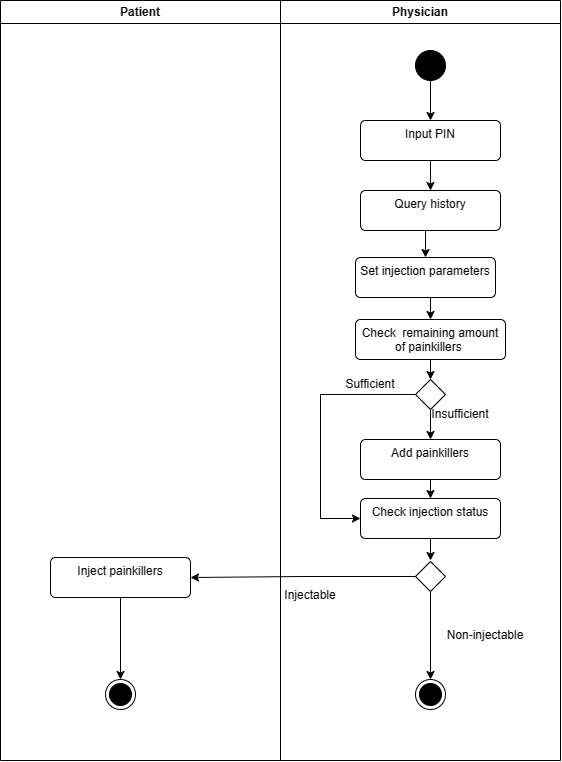
## Domain Analysis

Physicians and patients are the main object-oriented for painkiller injection systems.

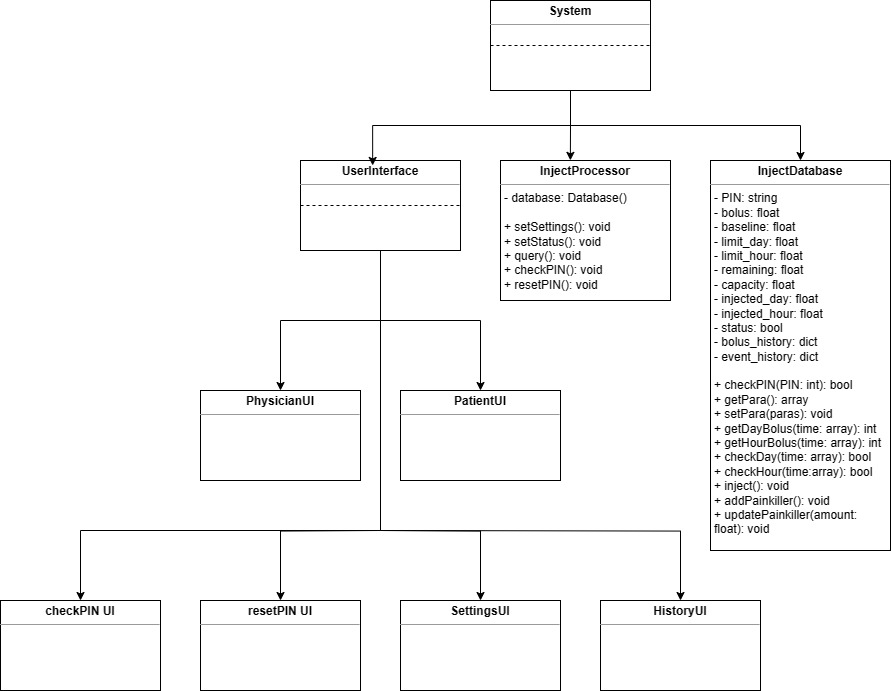
Here is the Class Diagram of them:



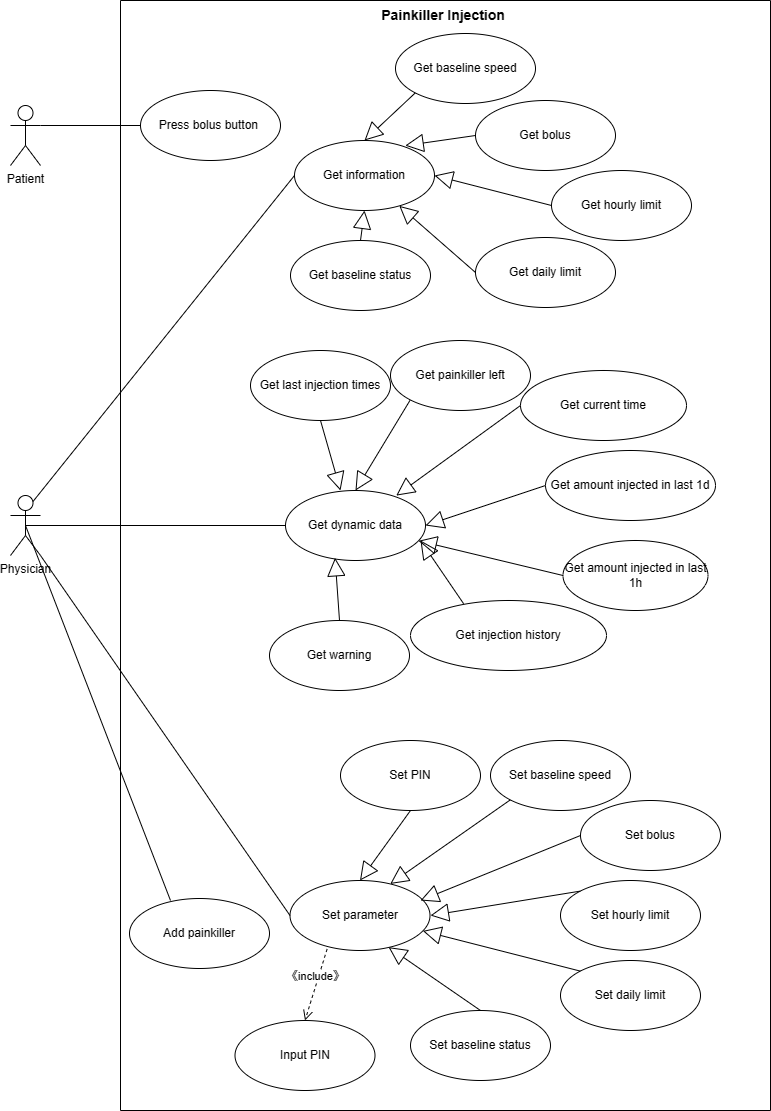
Here is the sequences of events for injecting painkiller:



## System Architecture

Painkiller injector system contains user interface, injection processor and injection database. 

## Use Cases

The system can accommodate the following use cases for both physicians and patients: 

## Software Requirements

### R1: PhysicianUI

* R1.1: The physician can set parameters
  + R1.1.1:The physician can set PIN
  + R1.1.2: The physician can set baseline speed
  + R1.1.3: The physician can set bolus
  + R1.1.4: The physician can set hourly limit
  + R1.1.5: The physician can set daily limit
  + R1.1.6: The physician can set baseline status
  + R1.1.7: The physician can add painkiller
* R1.2: The UI can show information
  + R1.2.1: The UI shows baseline speed
  + R1.2.2: The UI shows bolus
  + R1.2.3: The UI shows hourly limit
  + R1.2.4: The UI shows daily limit
  + R1.2.5: The UI shows baseline status
* R1.3: The UI can show dynamic data
  + R1.3.1: The UI shows painkillers left
  + R1.3.2: The UI shows last injection time
  + R1.3.3: The UI shows current time
  + R1.3.4: The UI shows amount injected in last 1 day
  + R1.3.5: The UI shows amount injected in last 1 hour
  + R1.3.6: The UI shows injection history
* R1.4: The UI shows running out warning for painkillers

### R2: PatientUI

### R2.1: The UI shows the button for bolus which allows the patient to inject

### R3: InjectProcessor

* R3.1: The InjectProcessor should check PIN
* R3.2: The InjectProcessor should inject baseline gradually if baseline status is “on”. Otherwise, it should reject injecting.
* R3.3: The InjectProcessor should stop injecting painkiller bolus when meeting hourly and daily limit

### R4: InjectDB

* R4.1: The database should save parameters
* R4.1.1:The database saves PIN
* R4.1.2: The database saves baseline
* R4.1.3: The database saves bolus
* R4.1.4: The database saves hourly limit
* R4.1.5: The database saves daily limit
* R4.2: The database should record injections
* R4.2.1: The database records baseline injection
* R4.2.2: The database records painkiller bolus injection
* R4.2.3: The database records the remaining amount of painkillers in the container.
* R4.3: The database should calculate injected volume
* R4.3.1: The database calculates the painkiller injected in last 1 day
* R4.3.2: The database calculates the painkiller injected in last 1 hour