System Objective

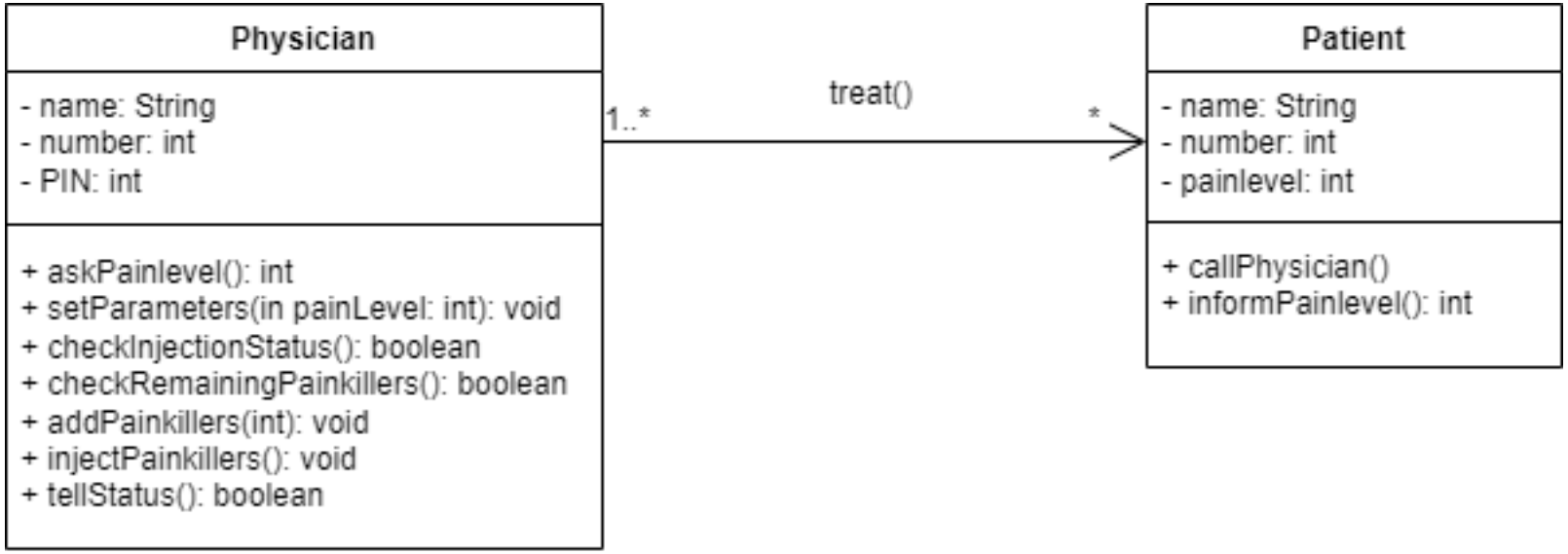
Introducing our Painkiller Injector--a revolutionary device designed to alleviate pain effectively and efficiently. With customizable dosage limits set by the attending physician, it automatically calculates and administers the appropriate painkillers to patients. Should the patient experience intense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

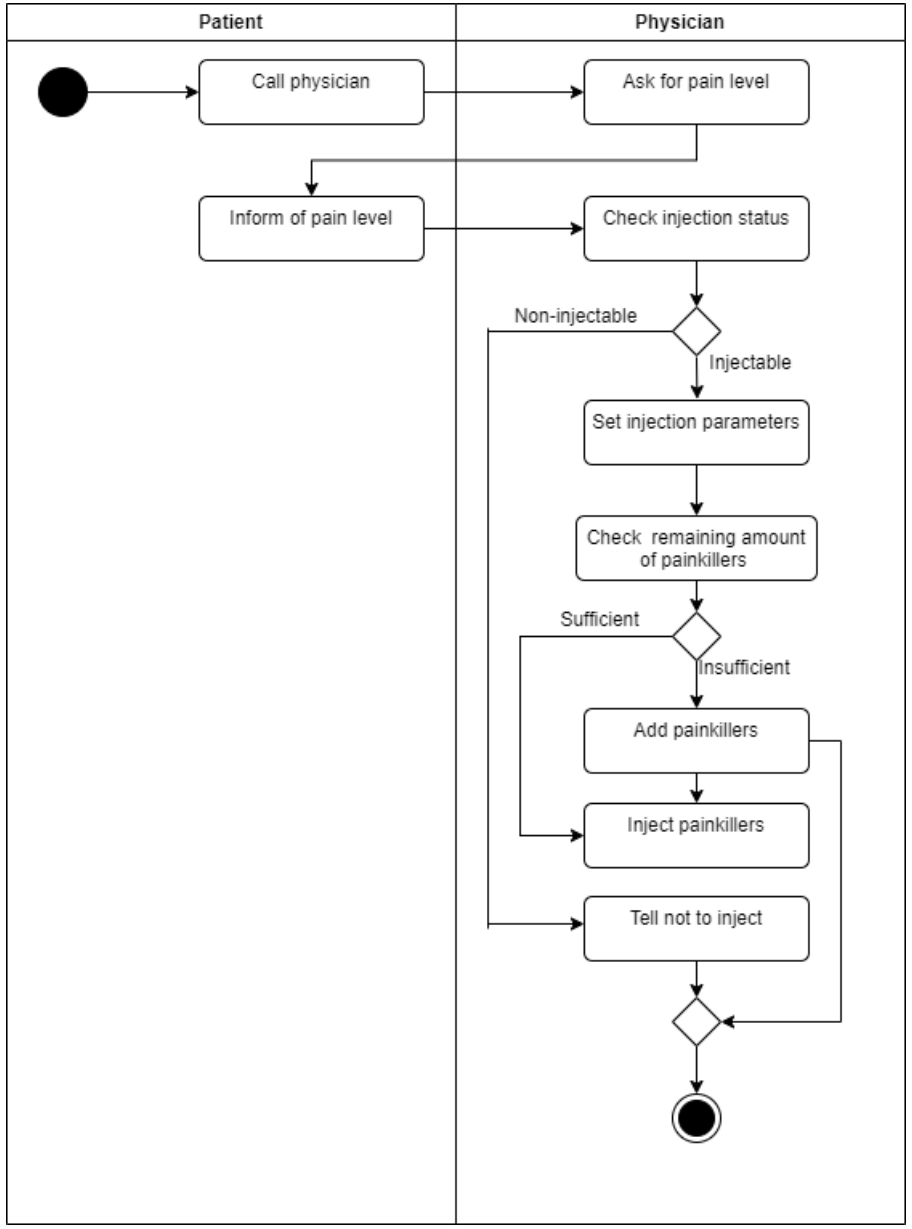
Domain Analysis

Class Diagram

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



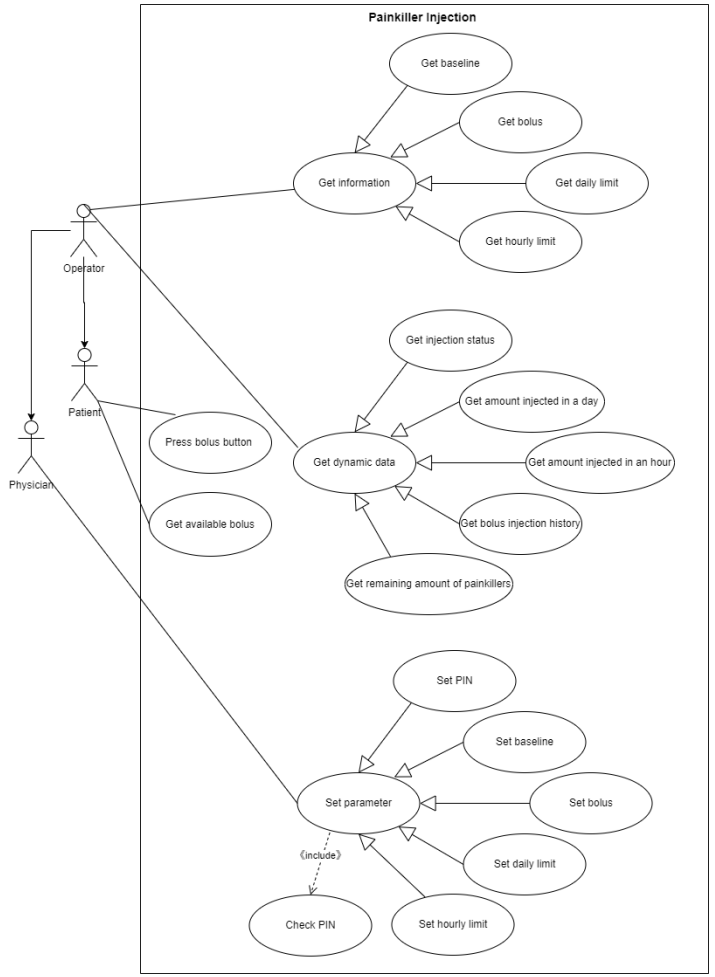
Activity diagram



System Requirements

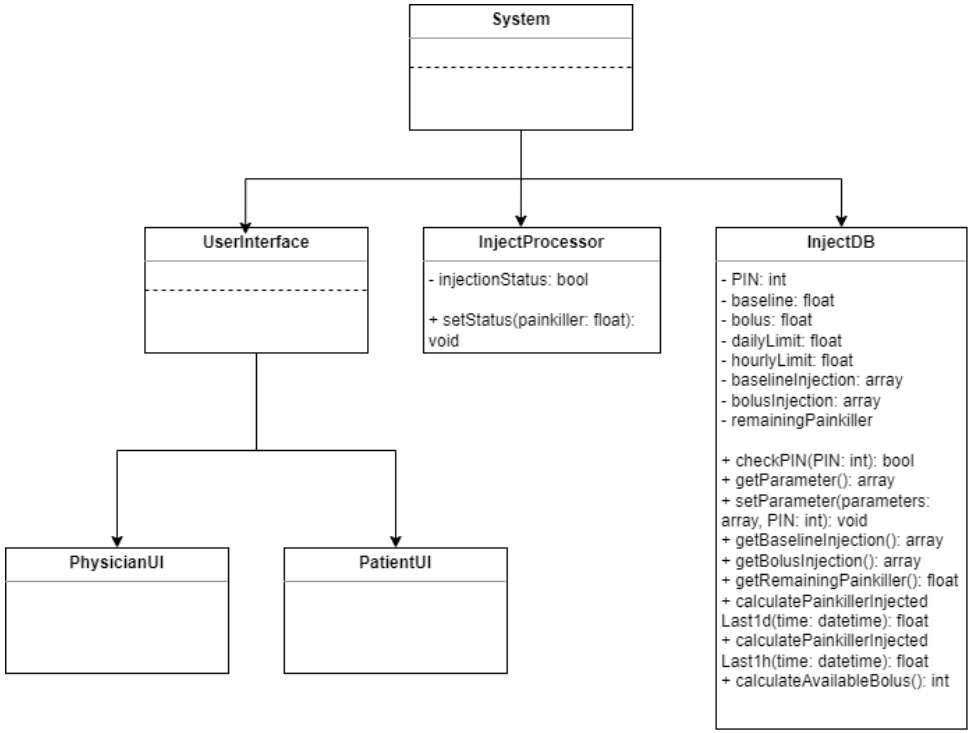
Use Cases

Users of painkiller injector system are primarily physicians and patients.



System Architecture

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



Text

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
  - R1.1.3: The physician can set bolus
  - R1.1.4: The physician can set daily limit
  - R1.1.5: The physician can set hourly limit

- R1.2: The UI can show information
  - R1.2.1: The UI shows baseline
  - R1.2.2: The UI shows bolus
  - R1.2.3: The UI shows daily limit
  - R1.2.4: The UI shows hourly limit
- R1.3: The UI can show dynamic data
  - R1.3.1: The UI shows injection status
  - R1.3.2: The UI shows amount injected in a day
  - R1.3.3: The UI shows amount injected in an hour
  - R1.3.4: The UI shows bolus injection history
  - R1.3.5: The UI shows remaining amount of painkillers
- R1.4: The UI shows low residual warning for painkillers

R2: PatientUI

- R2.1: The patient can press the button for bolus
- R2.2: The UI shows available bolus in a day
- R2.3: The UI shows low residual warning for painkillers

R3: InjectProcessor

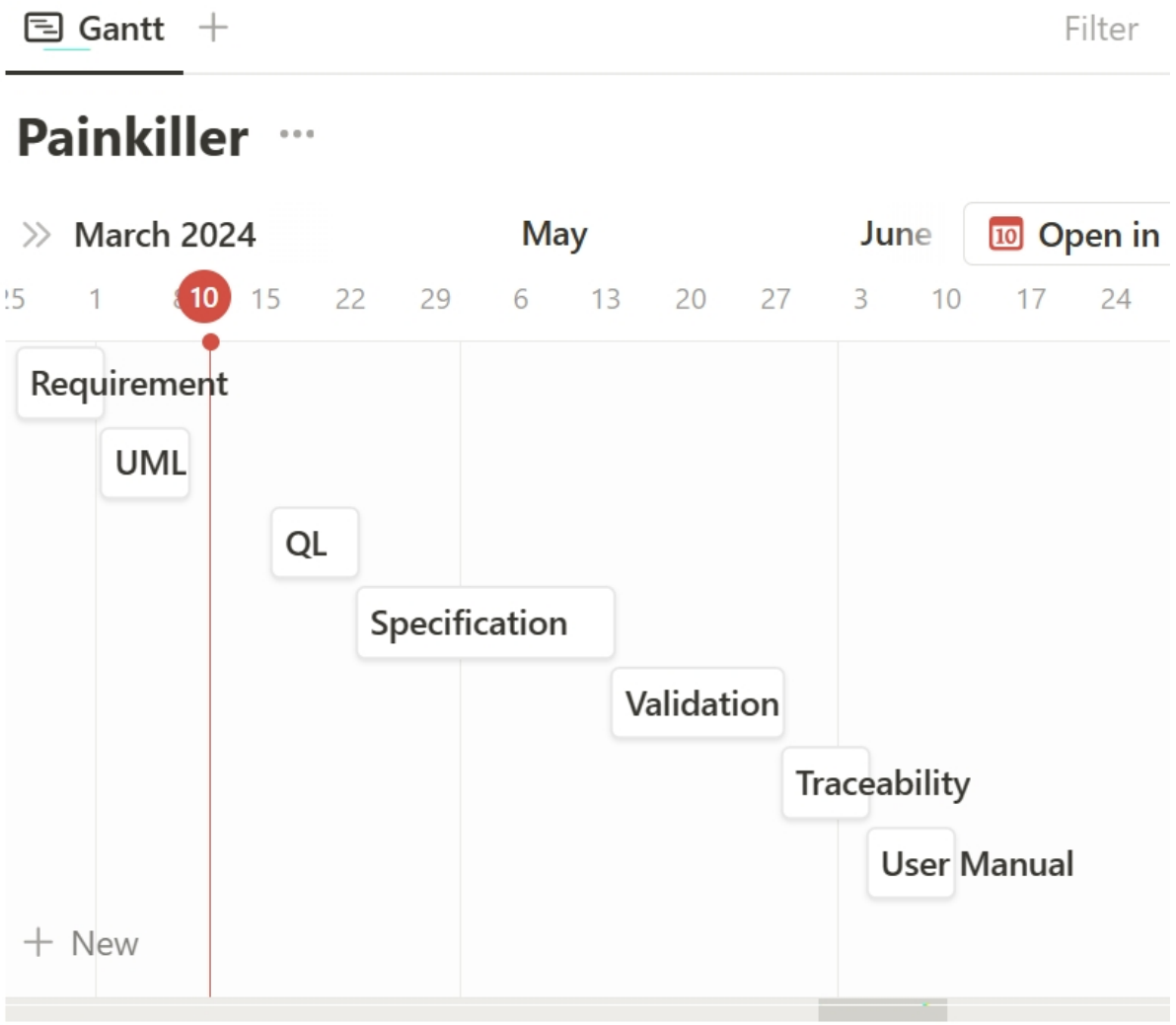
- R3.1: The injector should check PIN
- R3.2: The injector should inject painkiller gradually
- R3.3: The injector should stop injecting painkiller when meeting limit
  - R3.3.1: The injector should stop injecting painkiller when meeting daily limit
  - R3.3.2: The injector should stop injecting painkiller when meeting hourly limit
- R3.4: The injector should reject injecting painkiller if injecting bolus would exceed the limit
- R3.5 The injector should set injection status
  - R3.4.1: The injector sets "Injectable" within limit
  - R3.4.2: The injector sets "Non-injectable" when meeting limit or injecting would exceed the 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 daily limit
  - R4.1.5: The database saves hourly limit

- R4.2: The database should record injections
  - R4.2.1: The database records baseline injection
  - R4.2.2: The database records bolus injection
  - R4.2.3: The database records the remaining amount of painkillers from sensors
- R4.3: The database should calculate injected volume
  - R4.3.1: The database calculates the painkiller injected last 1 day
  - R4.3.2: The database calculates the painkiller injected last 1 hour
  - R4.3.3: The database calculates available bolus in a day

Timeline


Gantt
Filter
Painkiller
March 2024
May
June
Open in
25
1
8
10
15
22
29
6
13
20
27
3
10
17
24
Requirement
UML
QL
Specification
Validation
Traceability
User Manual
New

Questions

1. Class diagram and activity diagram are only used to help us with requirements analysis, not implementation, right? I think the specific implementation is written in the system architecture, and there are no doctors and patients in it.
2. Shouldn't the student in charge of the specification draw the system architecture?