Group 8

Author: Junda Shen

Software Requirements

Painkiller Injector

Table of Contents

[System Objective 2](#_Toc44468930)

[Domain Analysis 2](#_Toc44468931)

[System Architecture 5](#_Toc44468932)

[Use Cases 6](#_Toc44468933)

[Software Requirements 7](#_Toc44468934)

[R1: InjectorDB 7](#_Toc44468935)

[R2: InjectorProcess 7](#_Toc44468936)

[R3: InjectorUI 8](#_Toc44468937)

# System Objective

In this project, we are developing a software that can provide safe, convenient and accurate control of the painkiller injector system. By providing interconnected interfaces to injectors, physicians and patients, the system can reduce dangers during therapy and help patients receive more comfortable treatments, which can accelerate recovery processes for patients.

# Domain Analysis

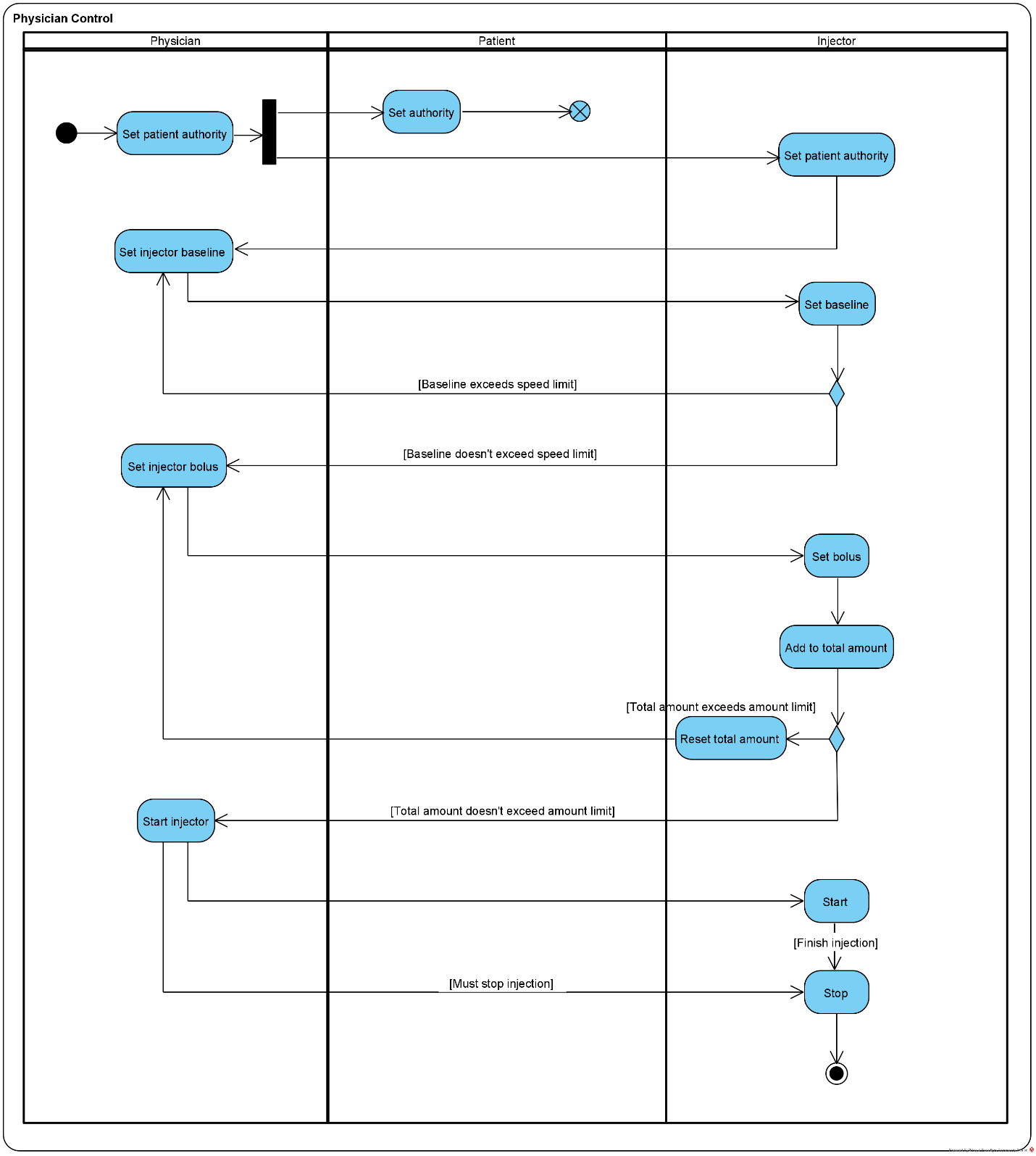
The participants of injection can be categorized into Patient, Physician and Injector.



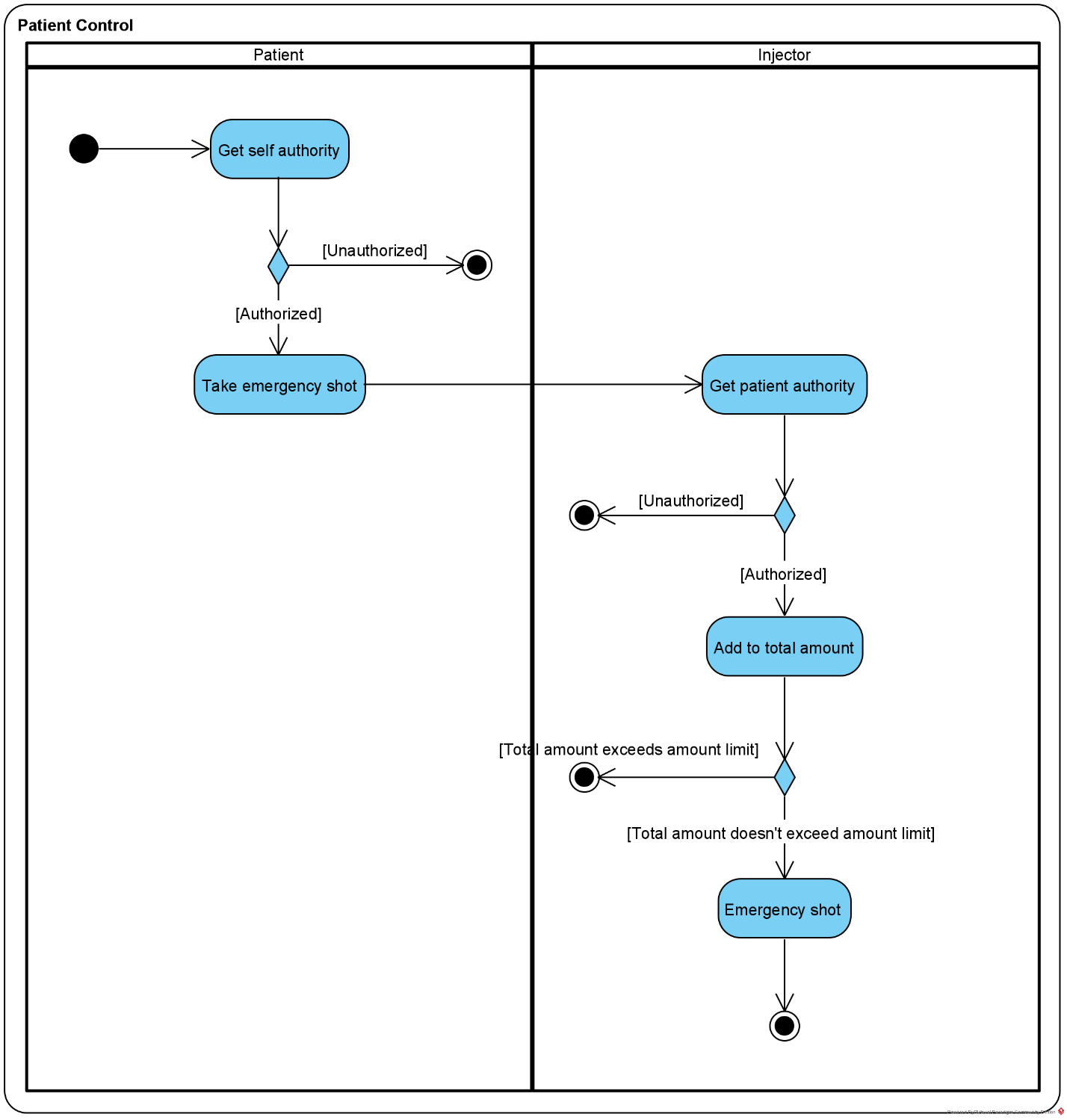
The relationships among different participants are shown as follows:



Here is the sequence of events for physician to control the injector:

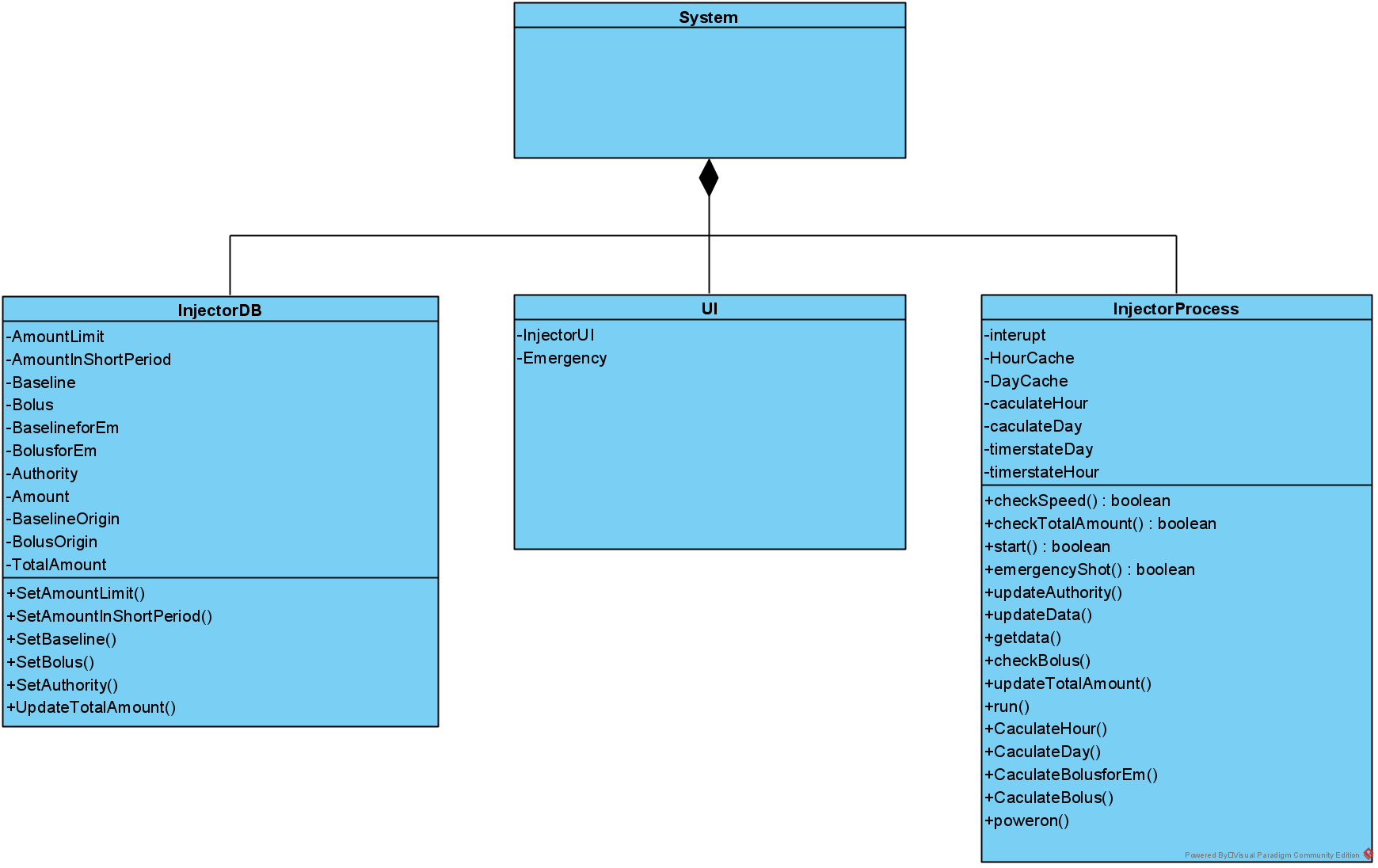


Here is the sequence of events for patient to control the injector:



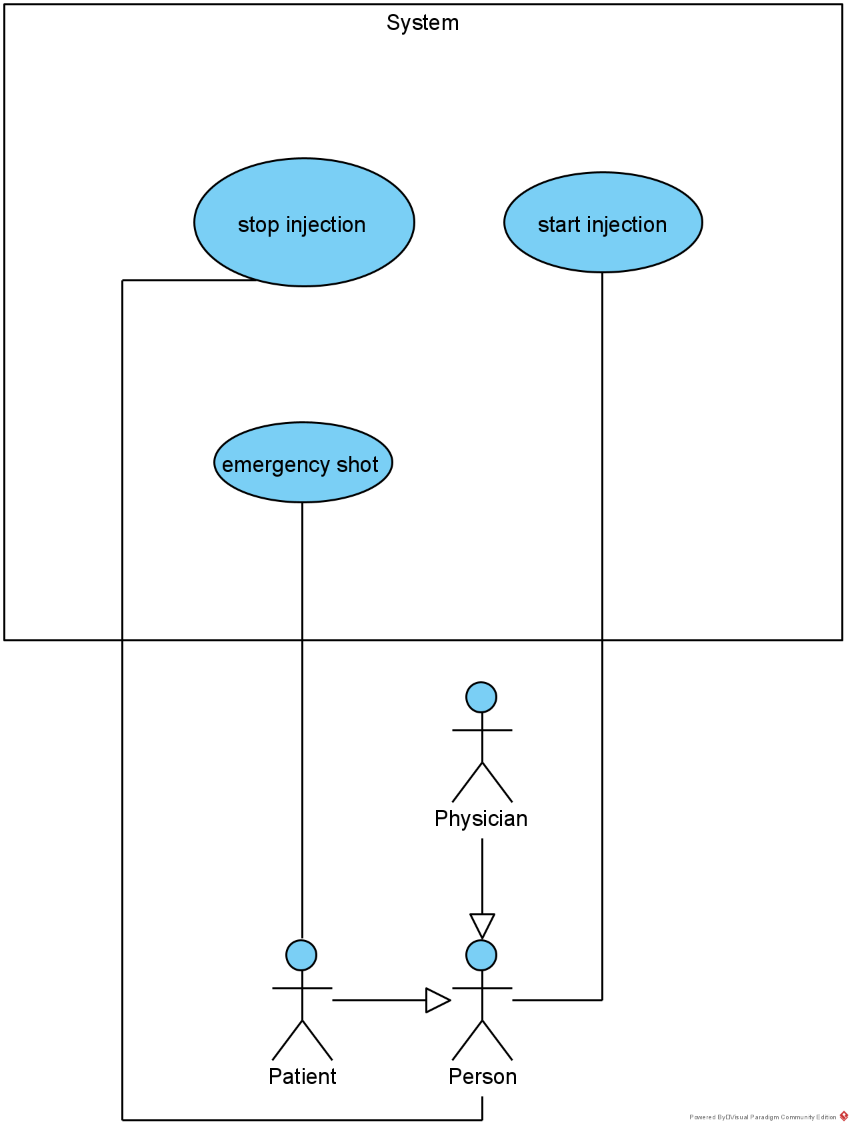
# System Architecture

From the information above, we will design a software system that allows the physician and patient to control the painkiller injector. The system architecture is shown below:



# Use Cases

The system can achieve the following use cases from the physician’s and the patient’s perspectives:



# Software Requirements

## R1: InjectorDB

* R1.1: The InjectorDB should be able to store and update data
  + R1.1.1: The InjectorDB should be able to store the amount limit and speed limit, which are built-in parameters
  + R1.1.2: The InjectorDB should be able to store baseline, baseline for emergency, bolus, bolus for emergency and patient authority
  + R1.1.3: The InjectorDB should be able to update and store the amount of painkiller that has been injected to the patient in a short period and in one day
* R1.2: The InjectorDB should be able to provide data to InjectorProcess

## R2: InjectorProcess

* R2.1: The injectorProcess should be able to check limits
  + R2.1.1: The injectorProcess should be able to check speed limit
  + R2.1.2: The injectorProcess should be able to check amount limit, both in short period and in one day
* R2.2: The injectorProcess should be able to control injection
  + R2.2.1: The injectorProcess should be able to start and stop injection
  + R2.2.2: The injector should be able to give emergency shot if the patient is authorized

## R3: InjectorUI

* R3.1: The injectorUI should be able to display current status
  + R3.1.1: The injectorUI should be able to display current speed of injection, amount had been injected and amount/time left
* R3.2: The injectorUI should be able to show deny information when some limits have been reached or internal exceptions happen
* R3.3: The injectorUI should be able to show input blocks, limits should be made on the input level first
* R3.4: The injectorUI should provide interface for emergency shot and authority check