Software Architecture

Medical Software Application

01/01/2017
LCIS
Cyril BRESCH

Table des matières

I. Introduction

This document provides all the modalities that have been realized under the development of a complete medical application simulator for internet of thing.

This document is organized as followed; first we introduce the project goals, the software and hardware resources, the planning and the risks. Then we will study the needs for a medical application simulator. The third part of this document is dedicated to the external specifications. The fourth part of the document deals with the conception phase of our simulator.

A - Project Goals

In order to make people understand security issues on medical devices, the LCIS has decided to create a basic medical application that will not implement specific securities. This medical application will be based on open RISC-V processor running both Linux and baremetal code. The whole platform is running on FPGA Nexys4 DDR. This FPGA represents an IoT pump that will be used in our next future in hospitals in order to cure patients. So our application will simulate the basic functioning of a medical injection pump. Then once this simulator is complete we will attack it to show that medical devices need dedicated security features.

B - Resources

In this project we have both hardware and software resources:

Hardware:

-FPGA Nexys 4 : Artix 7 processor, 100mHz, 16Mbyte of Cellular RAM, Micro SD access, UART access.

-FPGA Nexys 4 DDR:

Software:

- Xilinx Vivado : FPGA Flashing.

- Linux Ubuntu 14.04 in Virtual Machine.

- USB cables for the FPGA.

- RISC-V gcc, gdb tools.

- LowRisc toolchain.

C - Project lifecycle

In this project we decided to use an incremental lifecycle.

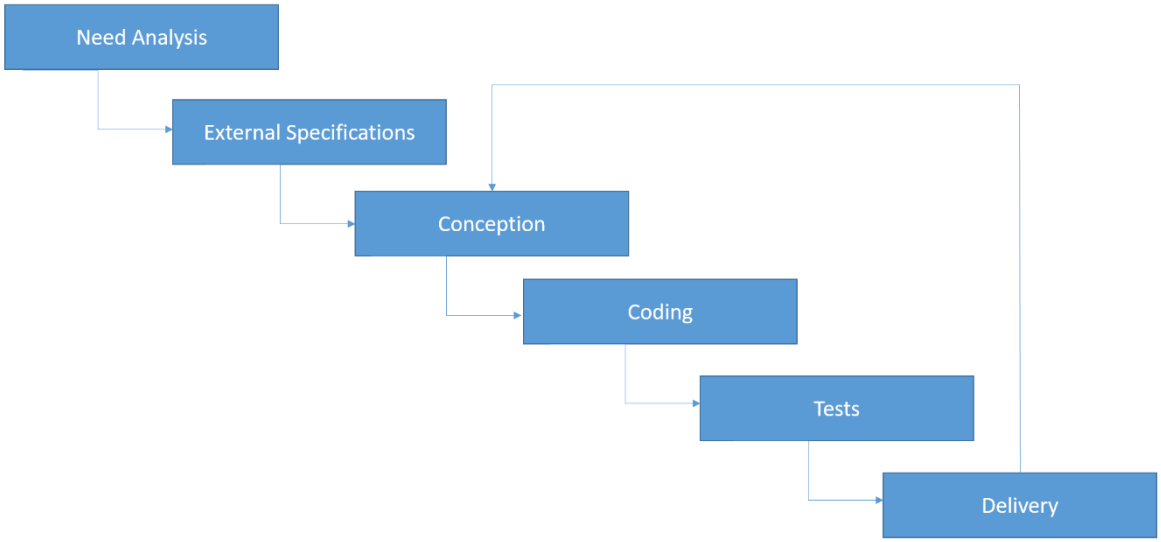


Figure 1 : Life cycle

During the specification phase all the cases of use of the different actors are covered and detailed. (see section 3)

D - Planning Supervision

E - Risks Management

In this project we can define three major types of risk; Technical risks, Human risks, Functional risk.

Risk Factor	Risk limitation
Technical Risk	
RISC-V toolchain and LowRISC evolution.	We concentrate on a specific stable version, LowRISC untethered 0.2 and priv-1.9 for the RISC-V debugger.
Toolchain complexity and buggy behavior of the LowRISC project.	We follow the LowRISC development mailing list to fix/avoid known bugs.
Lack of documentation about the LowRISC toolchain.	We tried to modify as little as possible the LowRISC and RISC-V toolchain.
Human Risk	
Be exceeded by the complexity of the tools.	
Functional Risk	
Time management.	Incremental management, and priority on important tasks.
Obey software boundary.	Clearly defining the development boundary of the demonstrator.

Technical Risk

The LowRISC project provides tools for an easy implementation of the RISC-V architecture on FPGA. These tools are preconized to be installed in Ubuntu version 14.04 LTS that is now deprecate. Nowadays the LowRISC project is at his version 0.4, but we know that the version 0.2 is more stable due to its seniority this is we choose to use it. We also choose to use the priv-1.9 toolchain for RISC-V debugging because the majority of the community approved the stability. Unless we cannot assure that these tools don't contain critical bugs that will alter the development of the simulator and its implementation on the FPGA.

The LowRISC project provides a RISC-V cross compiler, which use a specific version of shared library which is called newlib. Newlib is a C library which is dedicated to the embedded systems, so the porting is less complex than the Libc. Therefore, less syscalls can be used by the developer, which decrease the complexity of the software development. In the case of the Linux kernel based simulator the Libc can be used rather than Bare-metal, so a more complex version of the simulator can be ported.

Human Risk

Porting a software that simulates the behavior of a medical application on a FPGA requires complex tools. In baremetal development we need to use the cross compiler provided by the lowRISC project/RISC-V toolchain, then we generate the bitstream (signal that will flash the cards) in order to see the code execution on a remote terminal (UART connection with the FPGA). These tools are incredibly complex and can have buggy behavior that can't be quickly fixed by the developer. Furthermore, the implementation of the Linux kernel increases the amount of bugs because we

increase the quantity of code on the platform. In order to reduce the risk of facing new critical bugs the developer follows the mailing list in order to be aware of the evolution of the project.

Functional Risk

This project duration is short, in two months we will provide a complete simulator of a medical application that will run both on Bare-metal and Linux / RISC-V FPGA platform. Then we will perform an intrusion test on the platform in order to show the danger of a non-secure medical application. Due to the short delay we need to focus on the major functionalities of such application and correctly define the boundary of the project.

II. Need analysis

A. Definitions and acronyms

Terms	Definitions
Bolus	It's a dose of drug that a patient can self-inject. Once a bolus injection is performed the patient must wait a certain period of time before performing a second injection.

B. Product Definition

A medical drug pump injection is a device that can be used in order to apply a cure to a patient. With the raise of the IoT, medical pumps tend to be more and more connected to the network. However, they can be vulnerable to many types of attacks due to their lack of security implementation. Our application will be based on a FPGA and will be used to simulate a real drug pump injection. Thus both hardware and software pentesting will be performed on the platform.

i. Case of use

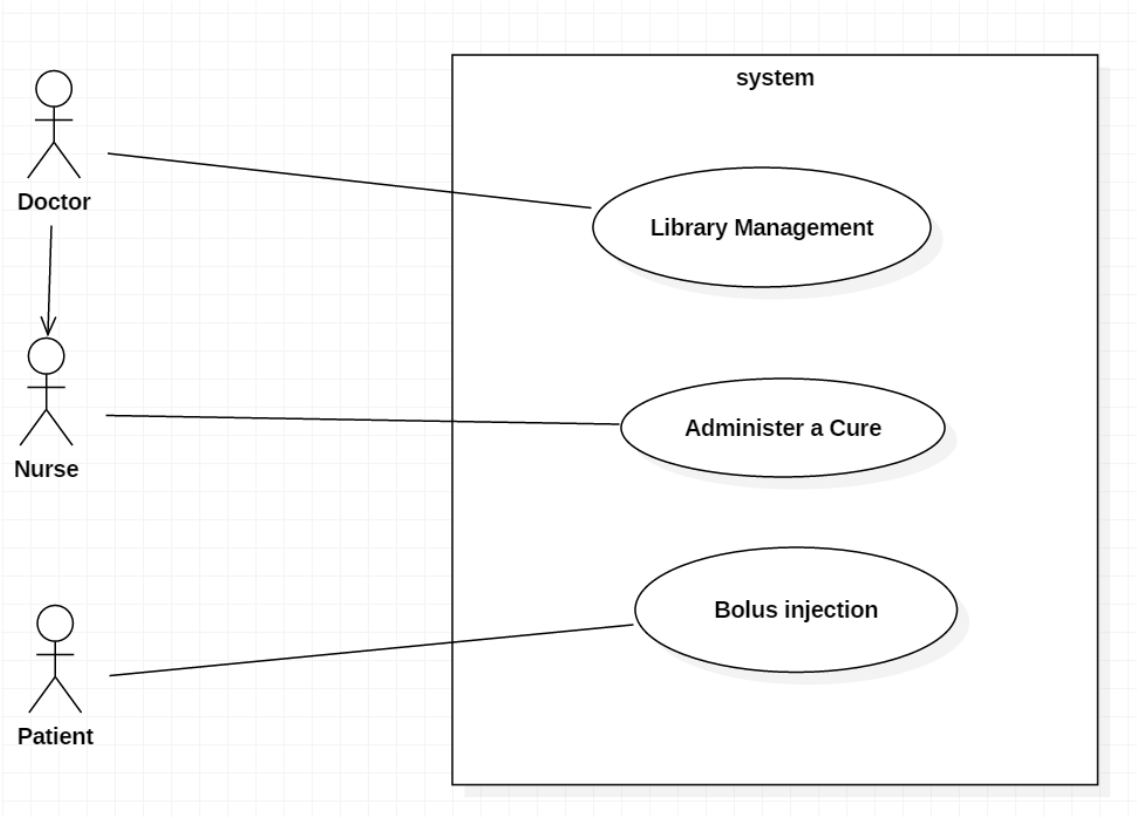


Figure 2 General Use Case Diagram

ii. Actors

In this project we have identified three major actors

The Doctor: This one can load new drugs in the library of the simulator, modify limit boundary of drug parameter, and delete drug from library. Doctors can also parameter cure for patient, including selecting drugs from library, launch/stop injection process, and finally modifying amount of drugs.

The Nurse: This one can parameter cure for patient, including selecting drugs from library, launch/stop injection process, and finally modifying amount of drugs.

The Patient: In this application patient unresponsive, the only they have to the pump simulator is the bolus injection function. They can self-inject a bolus periodically.

iii. Case of use description

ID	Task	Description
UC100	Library Management	-The administrator can upload new drugs in the Library. -The administrator can modify the boundary limit of drugs in the library. -The administrator can remove drugs from the

		Library.
UC200	Administer a Cure	-The user can select the drug to inject from library. -The user can parameter the amount of drug to inject. -The user can stop the injection process, then modify it. -The use can consult simulator Logs.
UC300	Bolus Injection	-The patient can self-inject a bolus. -After a bolus injection the system blocks this option during a predefined amount of time.

iv. Scenarios

Possible scenario for a Doctor

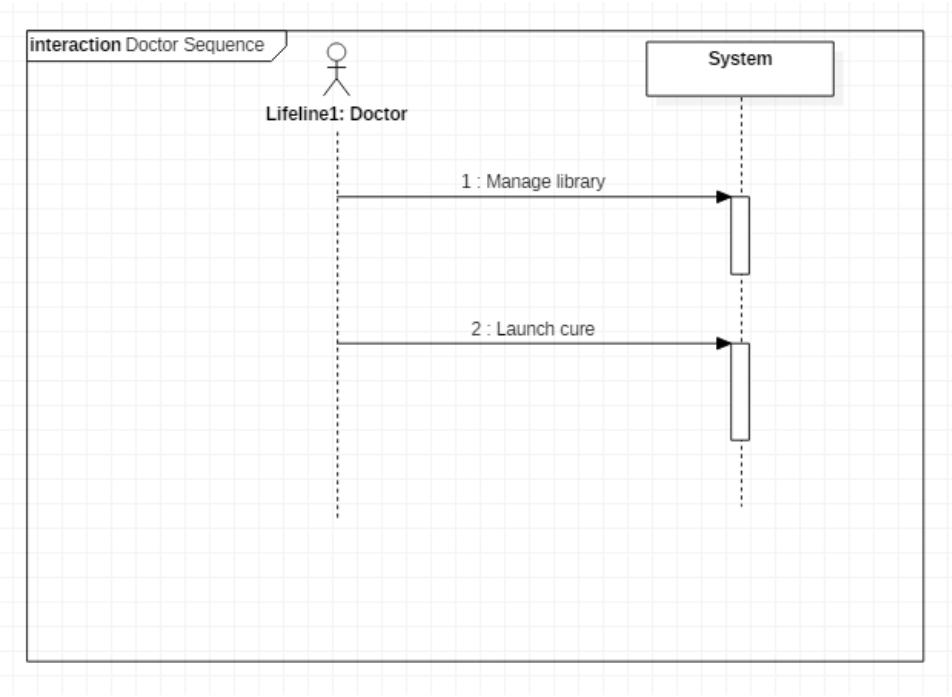


Figure 3 Doctor Scenario

Possible scenario for a Nurse

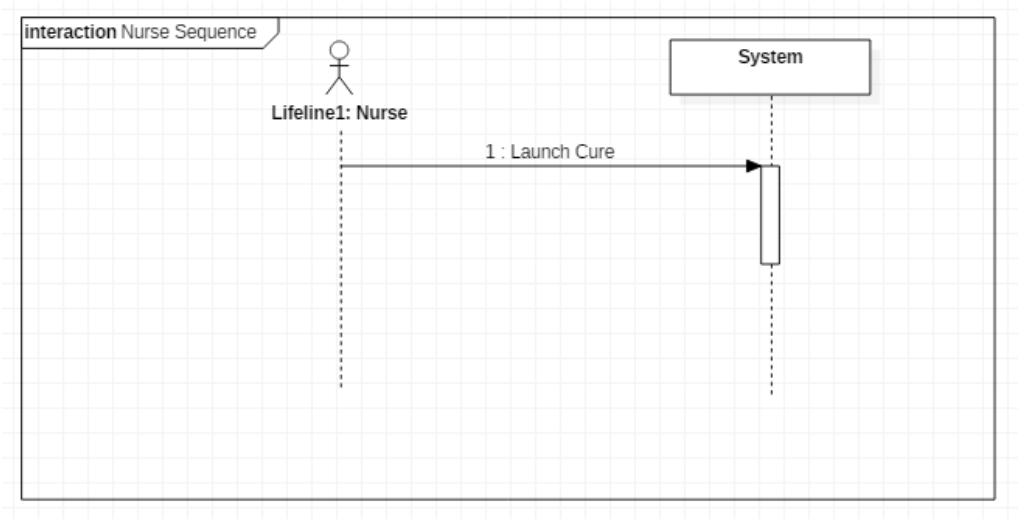


Figure 4 Nurse Scenario

C. Library Management

i. Use case diagram Library Management

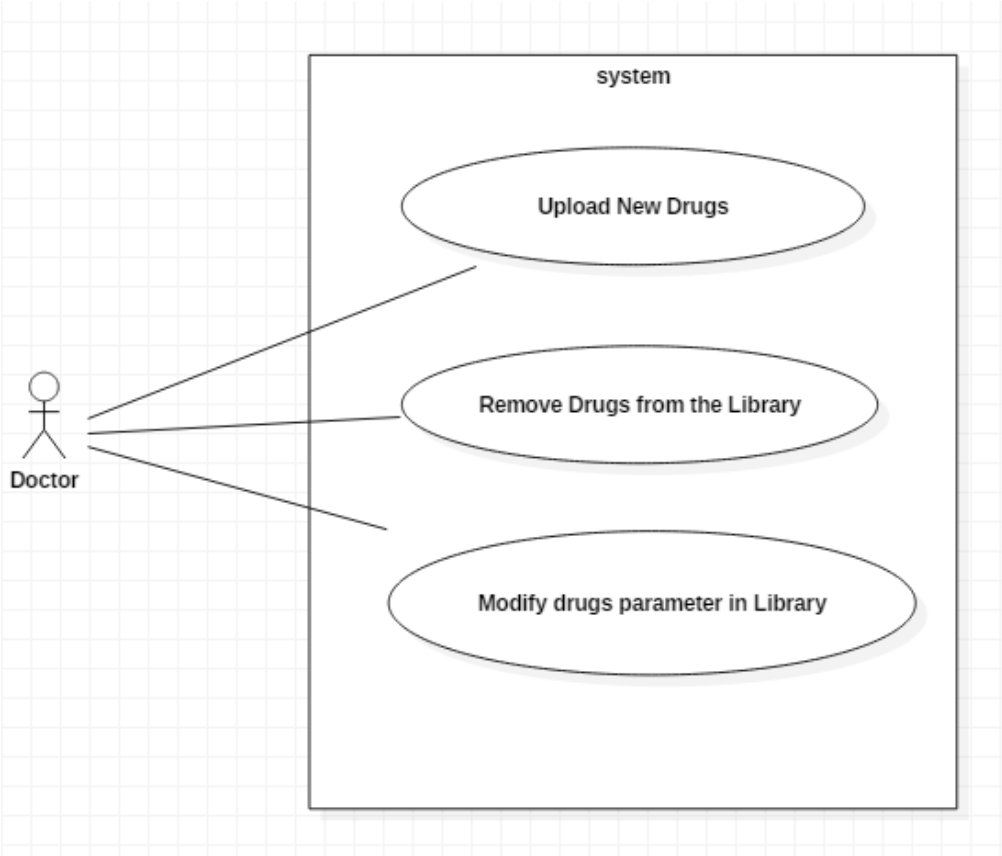


Figure 5 Library Management Use Case Diagram

ii. The actors

The Doctor: He can deploy new drugs in library, configure and remove them.

iii. Use case description

ID	Task	Description
UC101	Upload new drugs in Library	-The administrator (Doctor) can upload new drugs module into drugs library of the simulator. -Drugs modules come from software simulator provider -Once module is added the library is updated
UC102	Remove drugs from library	-The administrator can remove drugs from library of the simulator. -Once the drug module is removed the library is updated
UC103	Modify drugs from library	-The administrator can modify the default parameter of a drug module (limit of injection).

iv. Drug library Scenarios

Possible drug upload in the library scenario

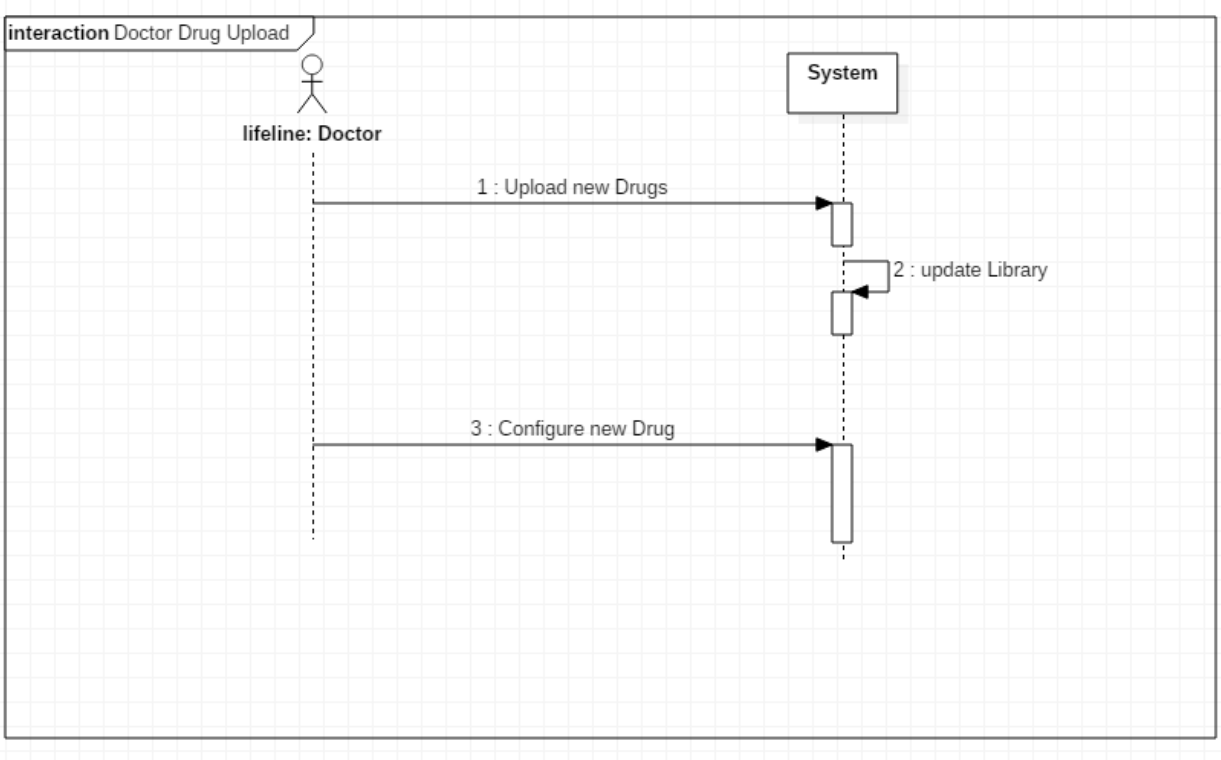


Figure 6 New Drug Upload

Drug configuration possible scenario

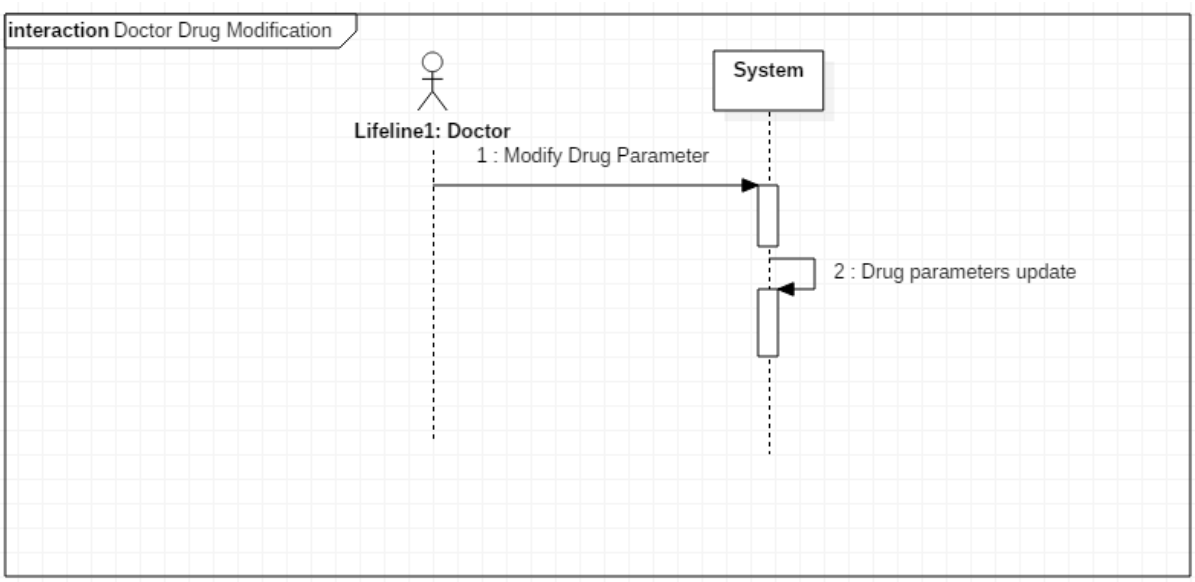


Figure 7 Drug configuration

Drug remove possible scenario

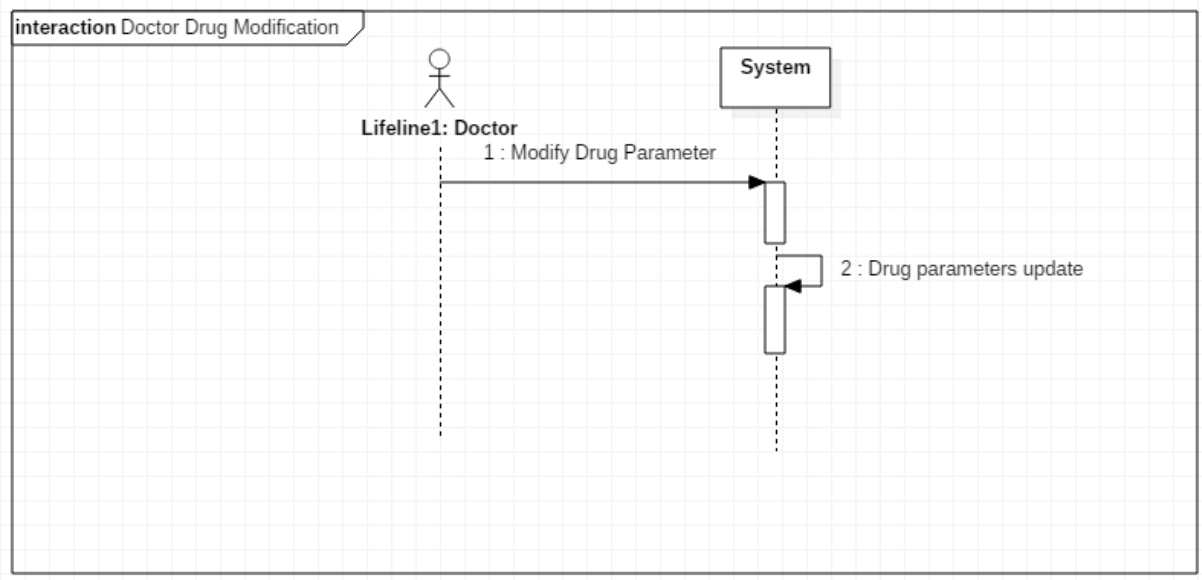


Figure 8 Drug modification

D. Cure Administration

i. Context Diagram

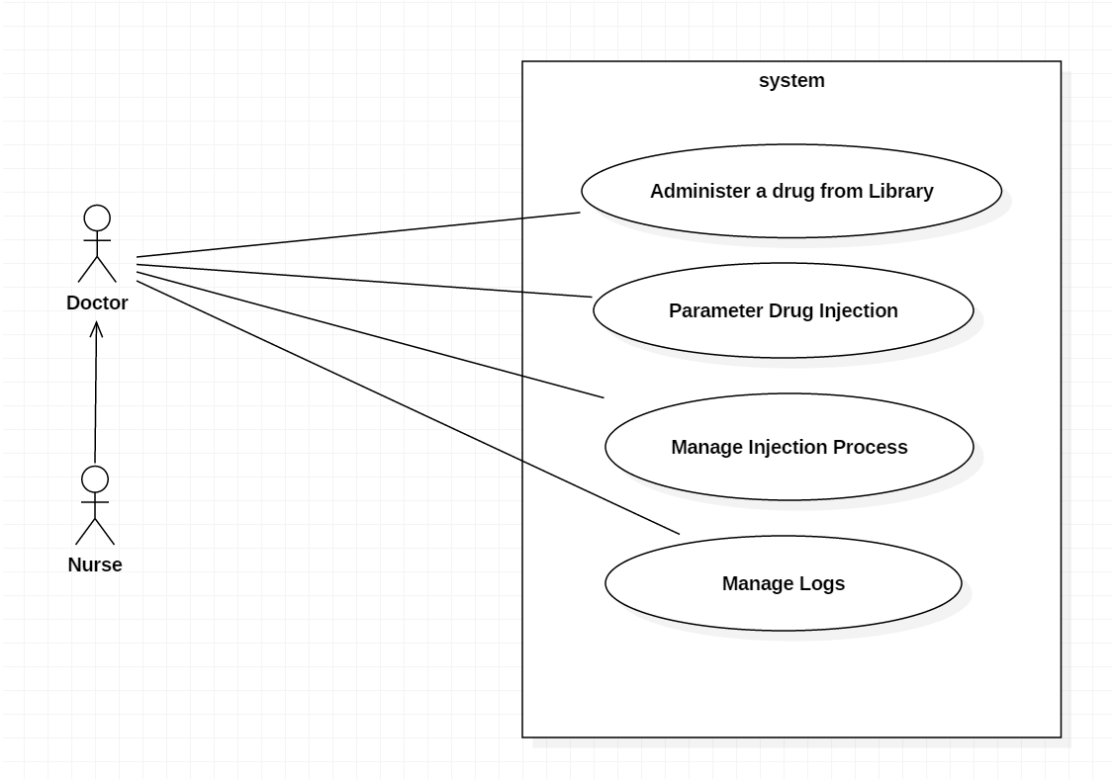


Figure 9 Cure Administration Use Case Diagram

ii. Actors

In this section we define the actors that can interact with the system (simulator) in order to administer a cure. The two actors are the Doctor and the Nurse, they have absolutely the same policy access

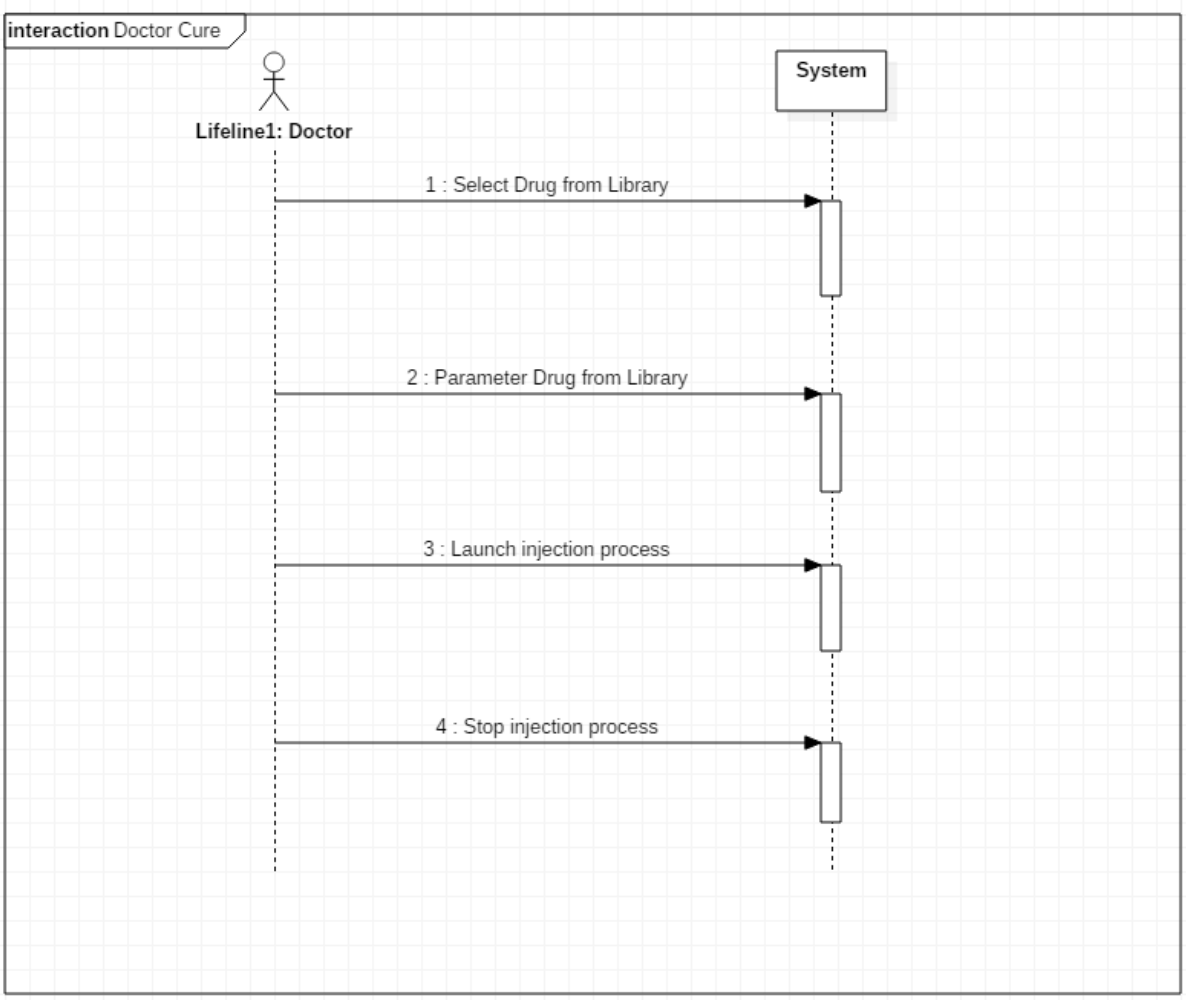
towards the system. They can administer a cure to a patient by loading a drug from the drug library. They can also stop the process injection and modify the amount of drug that will be injected for the patient. Finally, they can manage the logs of the pump.

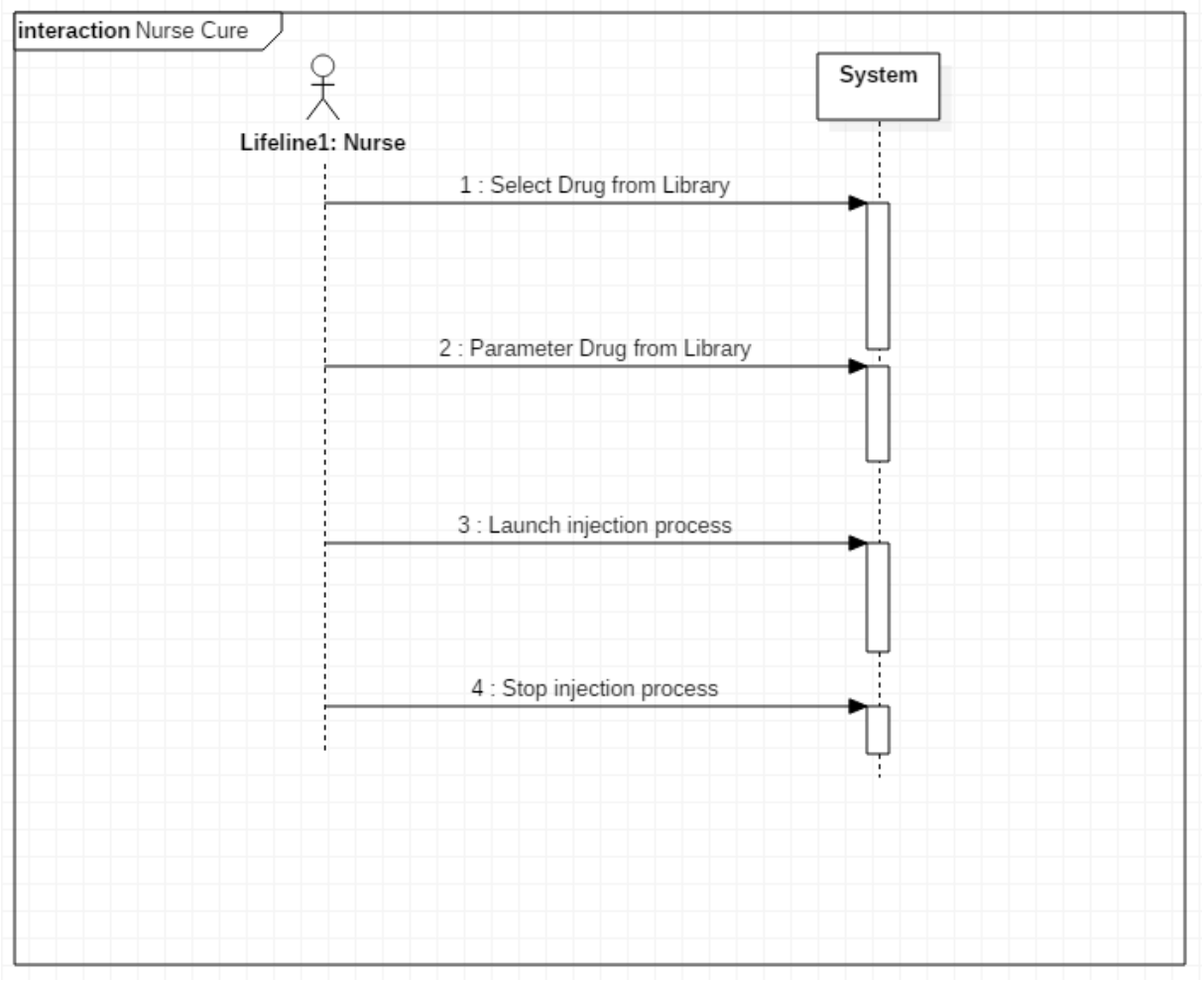
iii. Use Case Description

ID	Task	Description
UC201	Administer a drug from library	-Both administrator and user can choose a drug from the library and launch the injection process.
UC202	Parameter Drug Injection	-Once administrator or user have select the drug in library, they can modify, the default amount of drug to inject, the duration of the injection.
UC203	Manage Injection Process	-Both administrator and user can interrupt the injection process. It is possible to change the drug to inject, the amount of drug to inject, the duration of injection.
UC204	Manage Logs	-Both administrator and user can consult the logs of the pump. -They can delete logs from the pump.

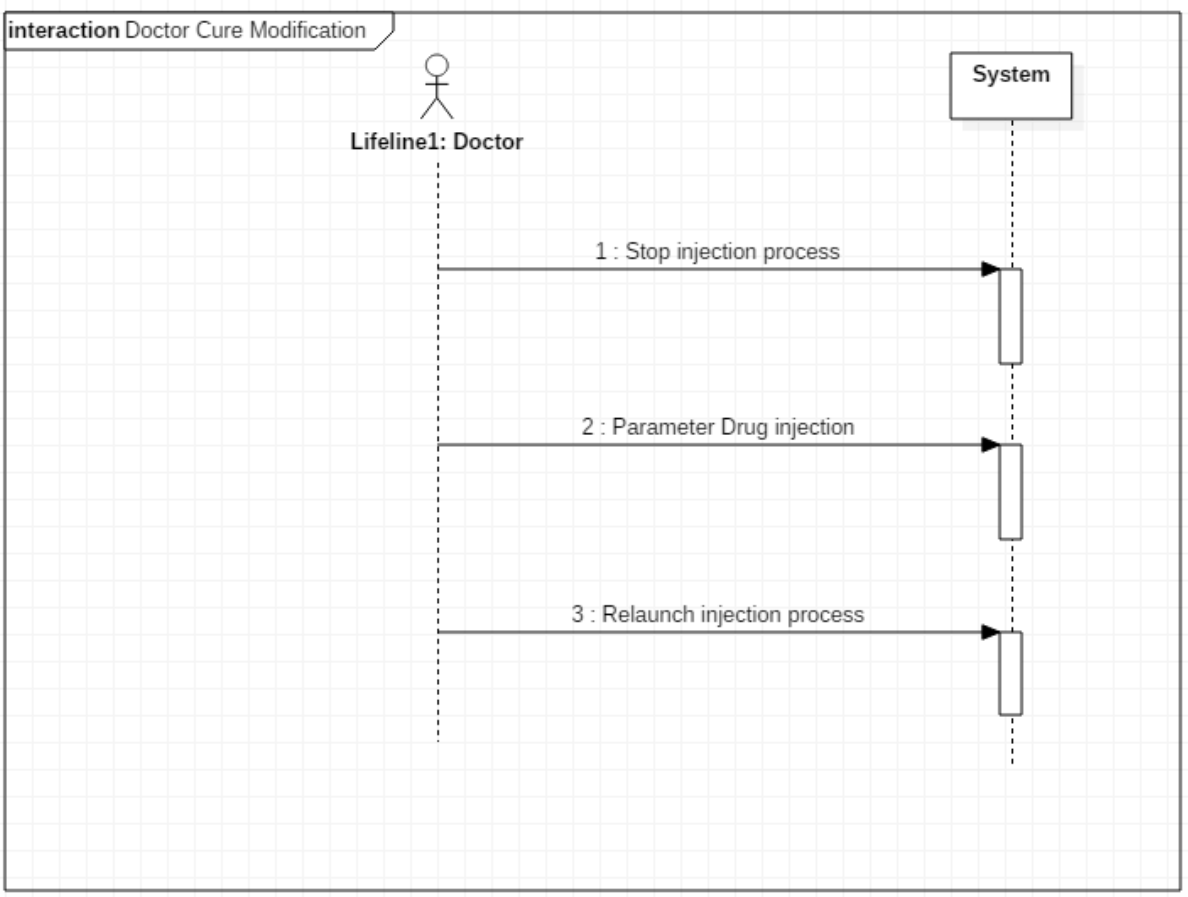
iv. Cure Management Scenarios

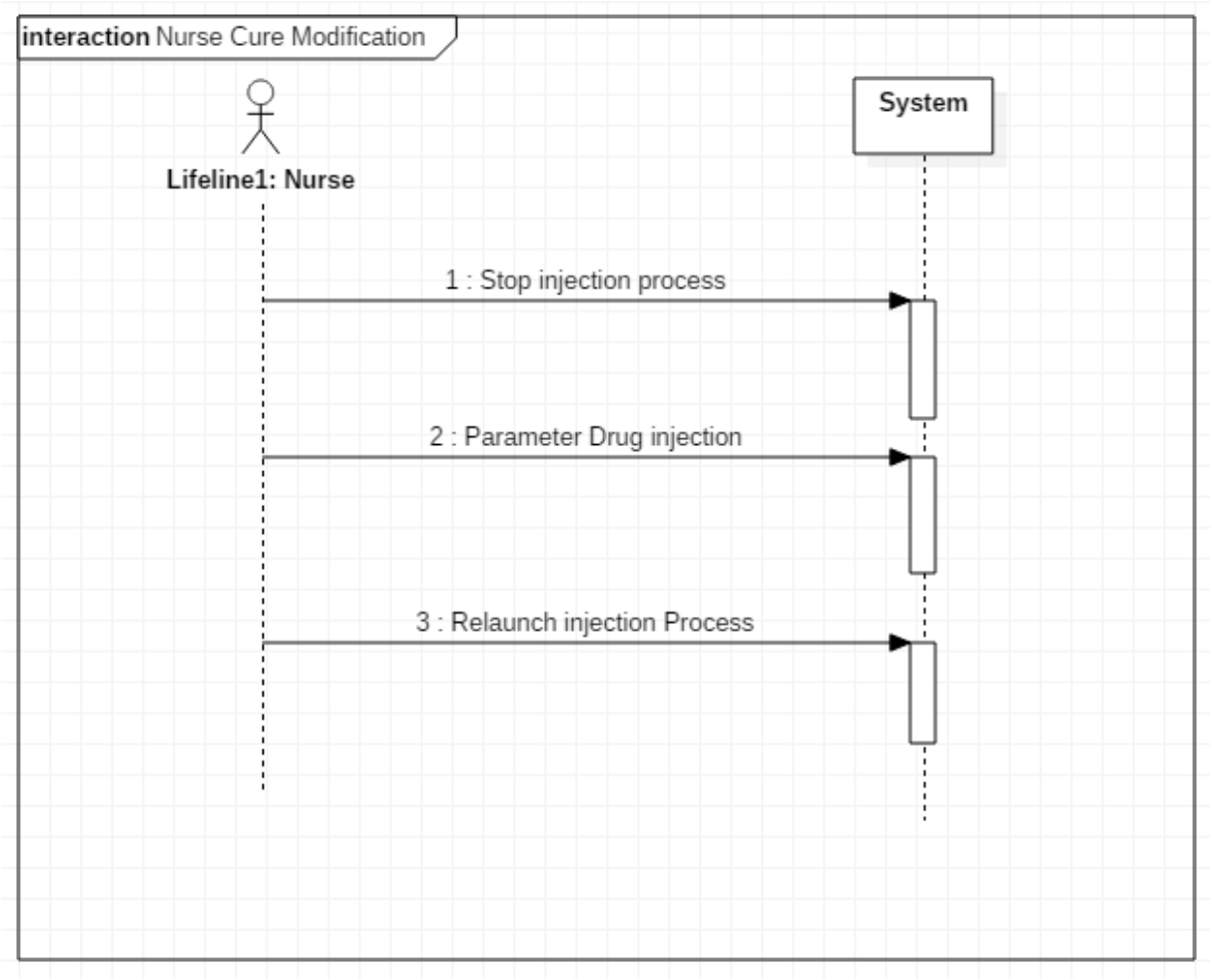
Possible scenarios for a cure administration


interaction Doctor Cure
Lifeline1: Doctor
System
1 : Select Drug from Library
2 : Parameter Drug from Library
3 : Launch injection process
4 : Stop injection process

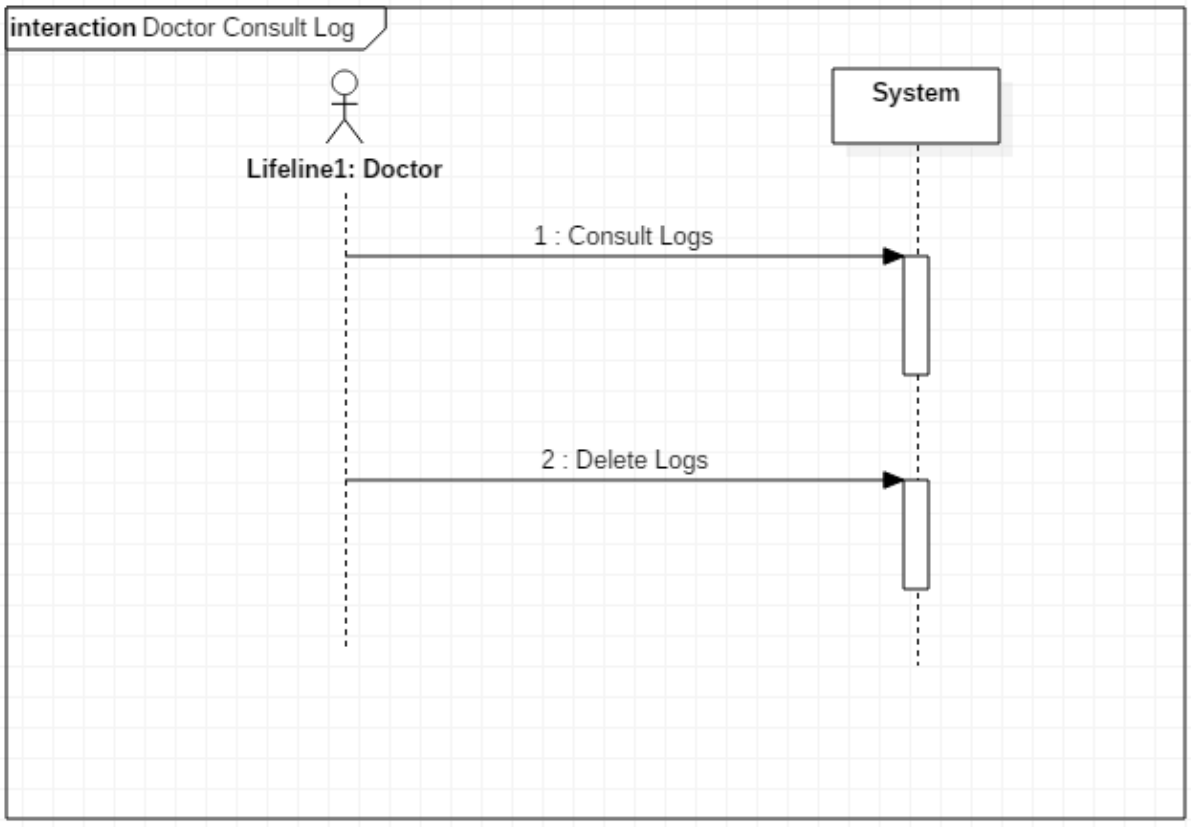


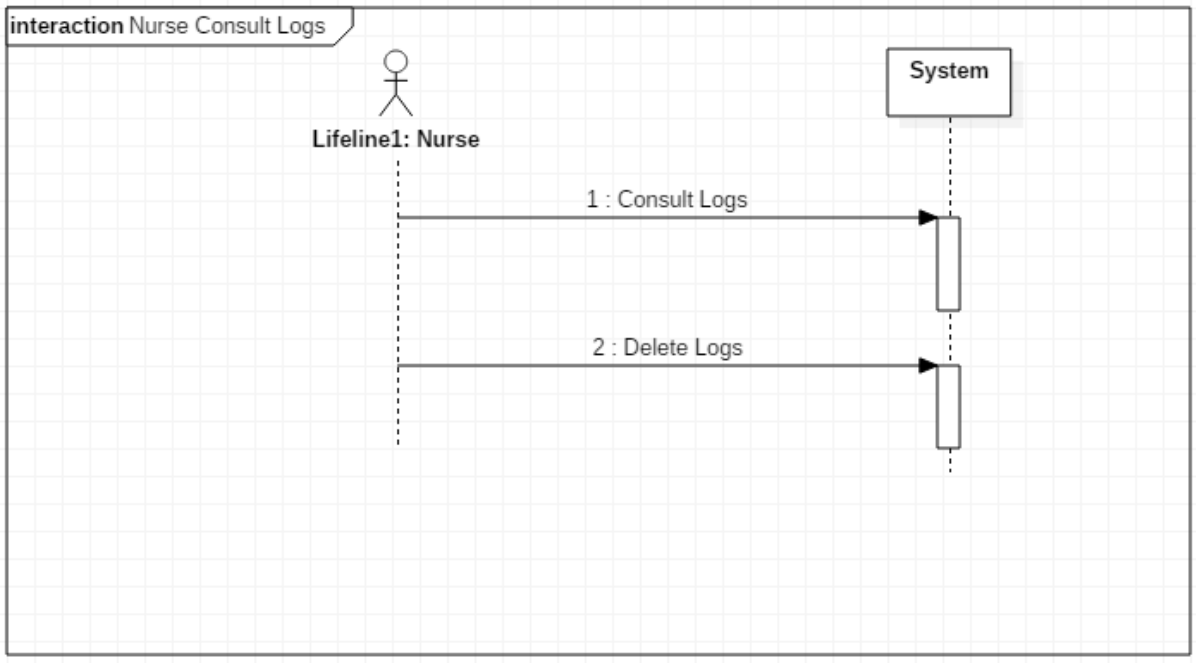
Possible scenario for cure modification


interaction Doctor Cure Modification
Lifeline1: Doctor
System
1 : Stop injection process
2 : Parameter Drug injection
3 : Relaunch injection process



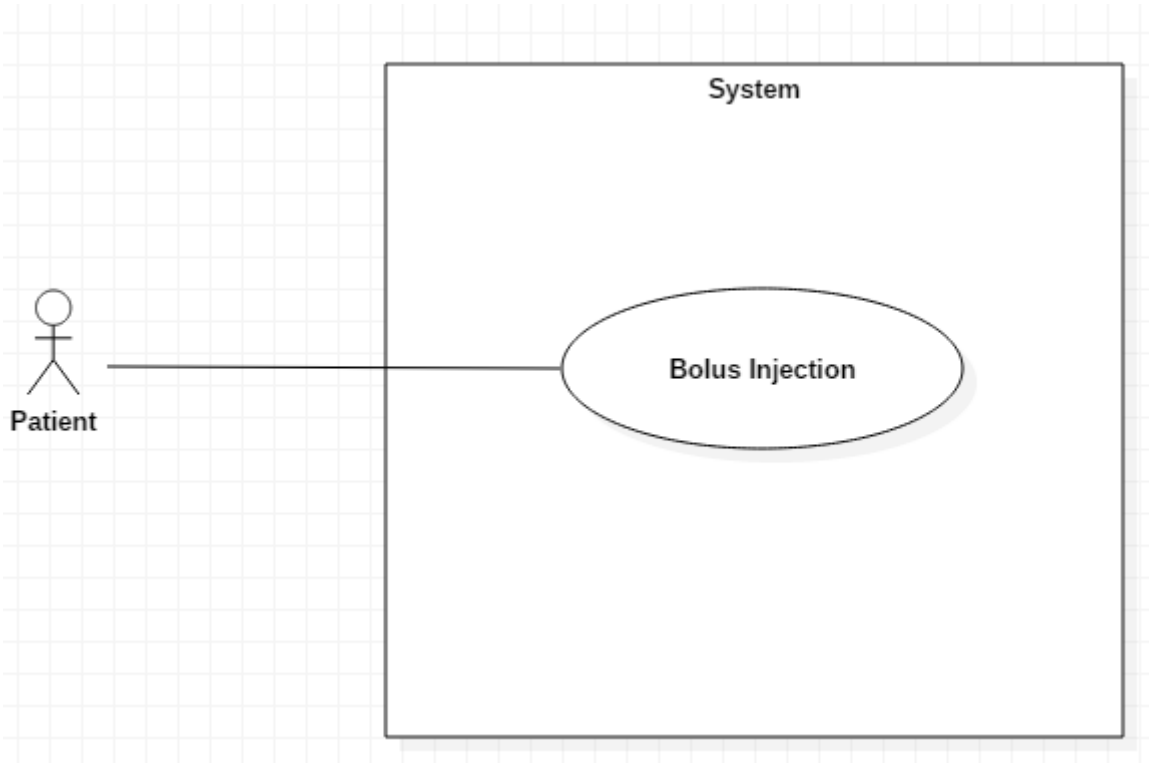
Consulting logs possible scenario


interaction Doctor Consult Log
System
Lifeline1: Doctor
1 : Consult Logs
2 : Delete Logs


interaction Nurse Consult Logs
System
Lifeline1: Nurse
1 : Consult Logs
2 : Delete Logs

E. Bolus Injection

i. Context Diagram



ii. Actors

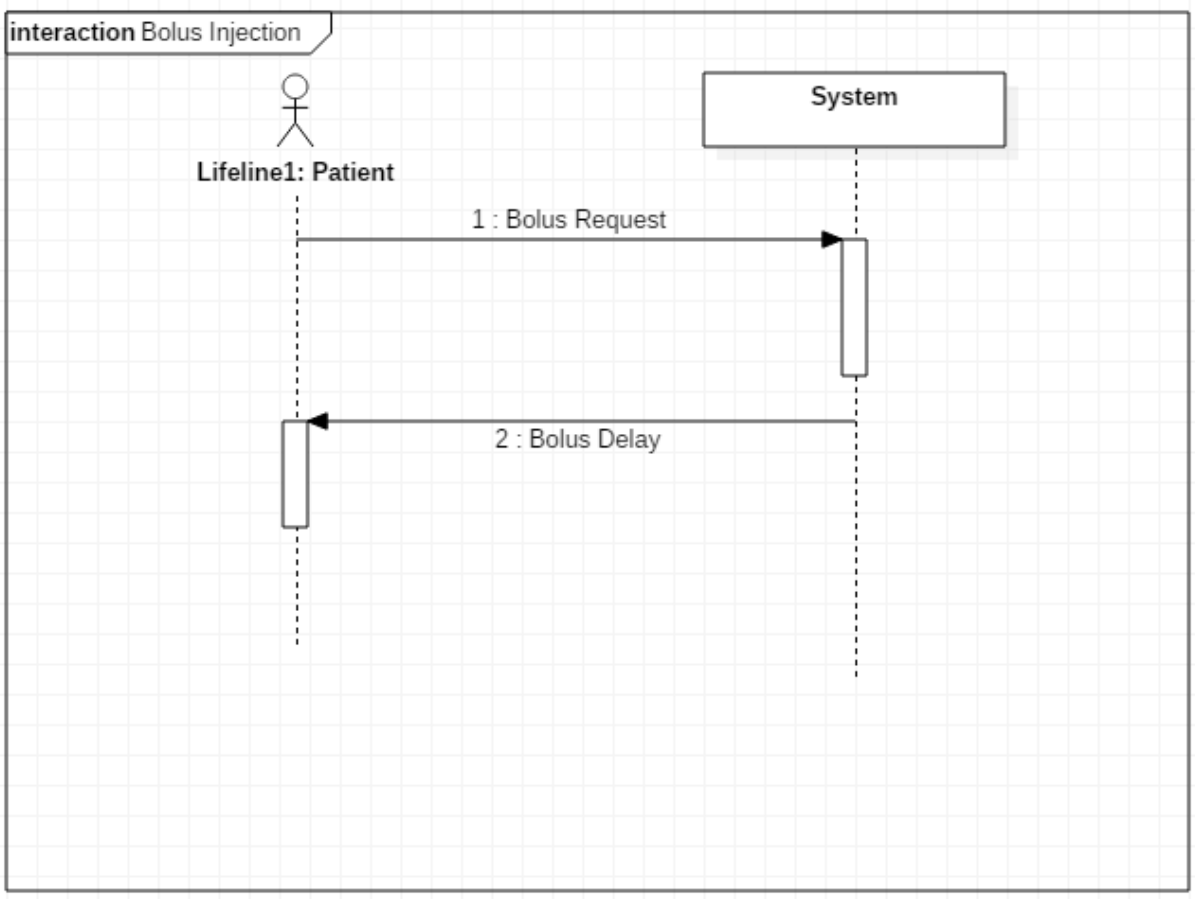
The Patient: During a cure a patient can self-inject a bolus. Once this injection is achieved, the patient must wait a certain period of time before requesting a new bolus.

iii. Use Case Description

ID	Task	Description
UC301	Bolus injection	-A patient can self-inject a bolus when this one is available by the simulator.

iv. Bolus Scenario

A possible scenario for bolus injection



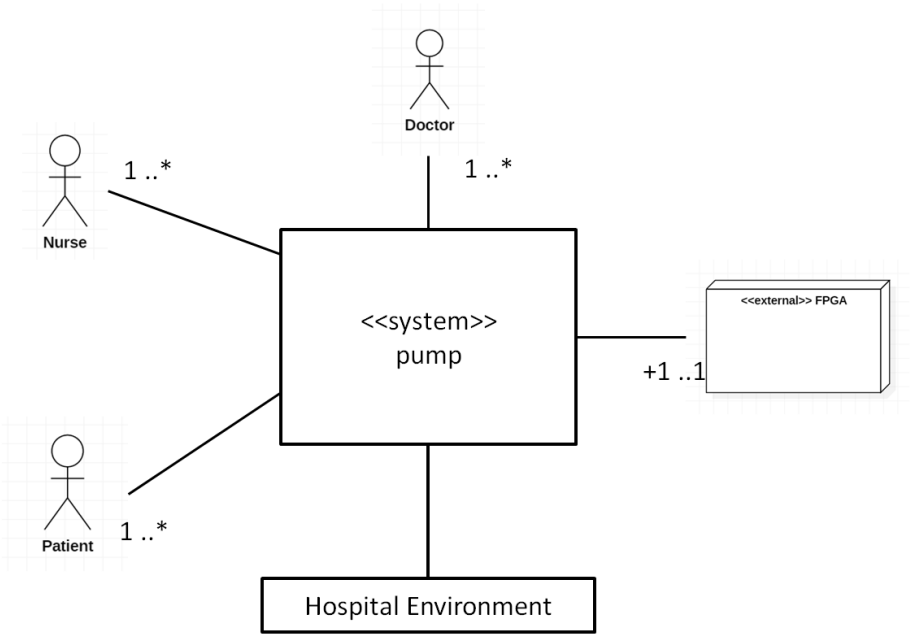
ID	Task	Description
UC100	**Library Management**	**-The administrator can upload new drugs in the Library.** **-The administrator can modify the boundary limit of drugs in the library.** **-The administrator can remove drugs from the Library.**
UC101	Upload new drugs in Library	-The administrator (Doctor) can upload new drugs module into drugs library of the simulator. -Drugs modules come from software simulator provider -Once module is added the library is updated
UC102	Remove drugs from library	-The administrator can remove drugs from library of the simulator. -Once the drug module is removed the library is updated

UC103	Modify drugs from library	-The administrator can modify the default parameter of a drug module (limit of injection).
UC200	**Administer a Cure**	**-The user can select the drug to inject from library.** **-The user can parameter the amount of drug to inject.** **-The user can stop the injection process, and then modify it.** **-The use can consult simulator Logs.**
UC201	Administer a drug from library	-Both administrator and user can choose a drug from the library and launch the injection process.
UC202	Parameter Drug Injection	-Once administrator or user have selected the drug in library, they can modify the default amount of drug to inject, the duration of the injection.
UC203	Manage Injection Process	-Both administrator and user can interrupt the injection process. It is possible to change the drug to inject, the amount of drug to inject, the duration of injection.
UC204	Manage Logs	-Both administrator and user can consult the logs of the pump. -They can delete logs from the pump.
UC300	**Bolus Injection**	**-The patient can self-inject a bolus.** **-After a bolus injection the system blocks this option during a predefined amount of time.**
UC301	Bolus injection	-A patient can self-inject a bolus when this one is available by the simulator.

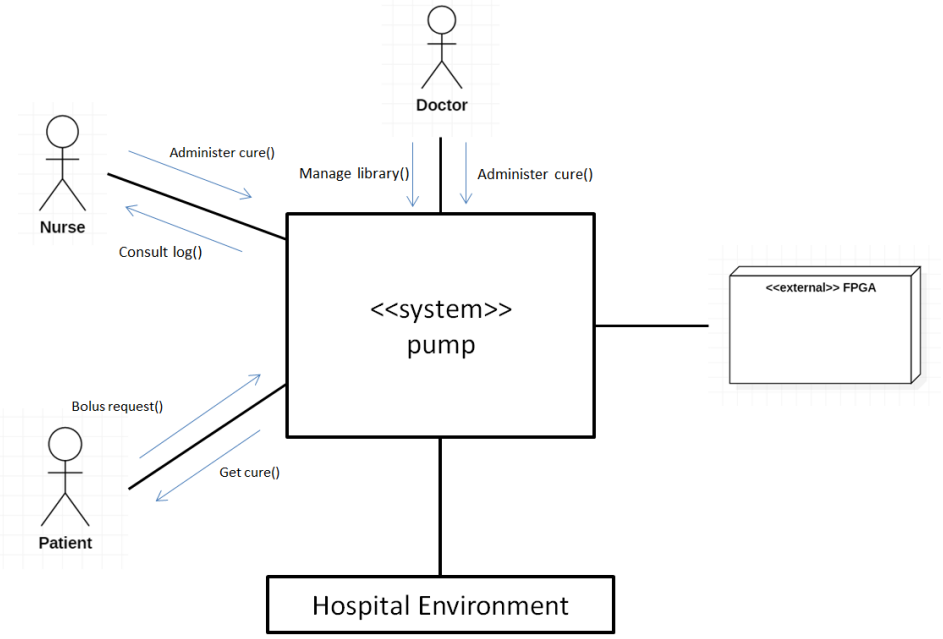
III. External specification

A. General Specification

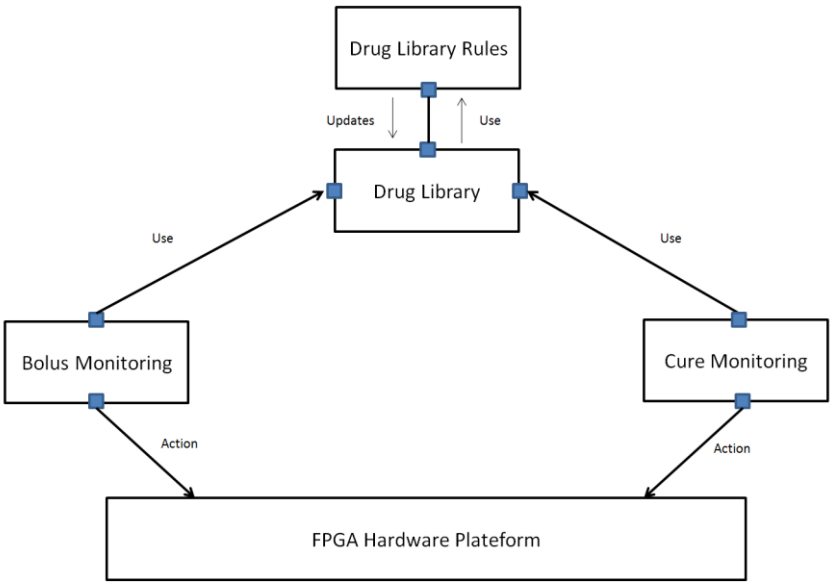
i. Static Context Diagram

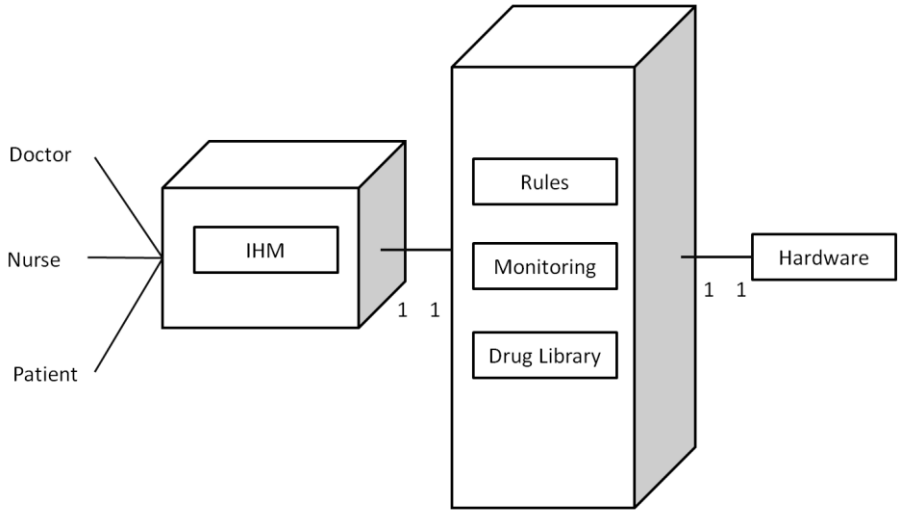


ii. Dynamic Context Diagram



iii. Logical and physical architecture view





B. Functional specifications

i. Library configuration

1. Update new drug in library

UC101 – 1 Configure Drug Name	
Goal	Adding a new drug name in the Library.
Actor	Doctor
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor on the Pump simulator. 2. Access to the library management interface.
Post Condition	The new drug name is added to the library.
Exception	

UC101 – 2 Configure Drug Quantity Limits	
Goal	Configuring Drug Limits in the Library
Actor	Doctor
Pre-Condition	To be logged on the IHM and have configure the drug Name.
Initial Scenario	1. Login as a Doctor on the Pump simulator. 2. Access to the library management interface. 3. Creating a new drug name.
Post Condition	The new drug limits are added to the library.
Exception	

2. Remove drug from library

UC102 Remove drug from library	
Goal	Remove a drug in the Library
Actor	Doctor
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor on the Pump simulator. 2. Access to the library management interface.
Post Condition	The drug will be removed from the library.
Exception	Removing drug when the library is empty.

3. Modify drugs from library

UC103 Modify drugs from library	
Goal	Modify a drug parameter from the library
Actor	Doctor
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor on the Pump simulator. 2. Access to the library management interface.
Post Condition	The drug selected will be modified from the library.
Exception	Changing drugs when the library is empty.

ii. Cure administration configuration

1. Cure Selecting

UC201 Selecting a drug from the library	
Goal	Selecting a drug to administer from the library
Actor	Doctor, Nurse
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor or Nurse on the Pump

	simulator. 2. Access to the drug administration interface.
Post Condition	The drug selected will be used for the cure.
Exception	

2. Drug configuration

UC202 Configure a drug from the Library	
Goal	Setting the amount of drug to inject
Actor	Doctor, Nurse
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor or Nurse on the Pump simulator. 2. Access to the drug administration interface.
Post Condition	The amount of drug selected will be configured for the cure.
Exception	The limited dose is checked by the application.

3. Drug Management

UC203 Manage Drug injection Process	
Goal	Modify the cure during operation
Actor	Doctor, Nurse
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor or Nurse on the Pump simulator. 2. Access to the drug administration interface. 3. A cure is launched.
Post Condition	The current injection process will be stopped
Exception	

4. Logs Management

UC204-1 Consult Logs	
Goal	Consult logs of the simulator
Actor	Doctor, Nurse
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor or Nurse on the Pump simulator. 2. Access to the Log interface.
Post Condition	Consult the Logs of the pump concerning a patient cure.
Exception	Trying to access patient data without pass.

UC204-2 Delete Logs	
Goal	Delete logs of a patient
Actor	Doctor, Nurse
Pre-Condition	To be logged on the IHM.
Initial Scenario	1. Login as a Doctor or Nurse on the Pump simulator. 2. Access to the Log interface.
Post Condition	Consult the Logs of the pump concerning a patient cure.
Exception	Trying to access patient data without pass.

iii. Bolus injection

UC301 Self Bolus Injection	
Goal	Self-inject a bolus
Actor	Patient
Pre-Condition	To get a cure from the pump
Initial Scenario	1. Getting a cure from the pump simulator.
Post Condition	A delay is set after the injection
Exception	Bolus injection before the delay.

C. Interface specification

i. Human Machine Interface

ii. Model

IV. Conception

A. Tasks

B. Test and validation