Software Architecture Project 1 Report

Health Centre Simulation

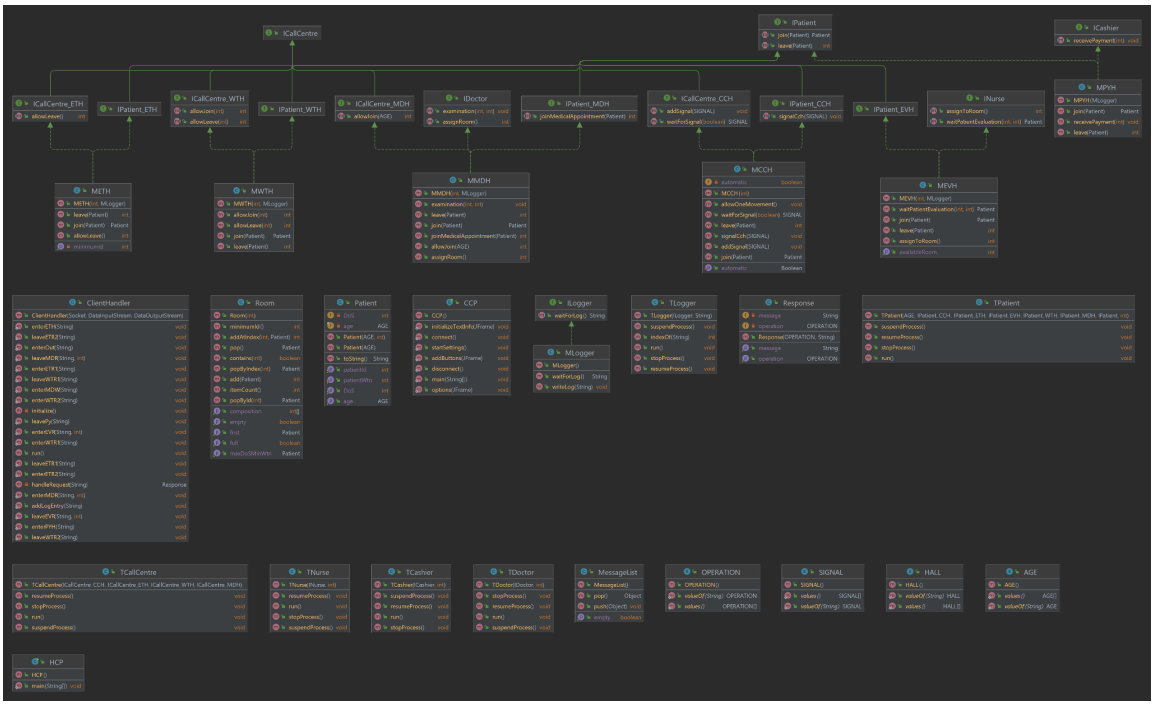
Pedro Marques, 92926 Inês Leite, 92928

Introduction

This report provides an overview of the project that we developed, related to the first practical assignment about the multithreaded Health Centre Simulation. Inside the delivered folder, there is a folder named JavaDocs which contains the documentation of every method and variable created in our project. Also, there are two folders, GUIs and StateDiagrams, with the several diagrams shown in this report, that we used to define how the different threads communicate and work together.

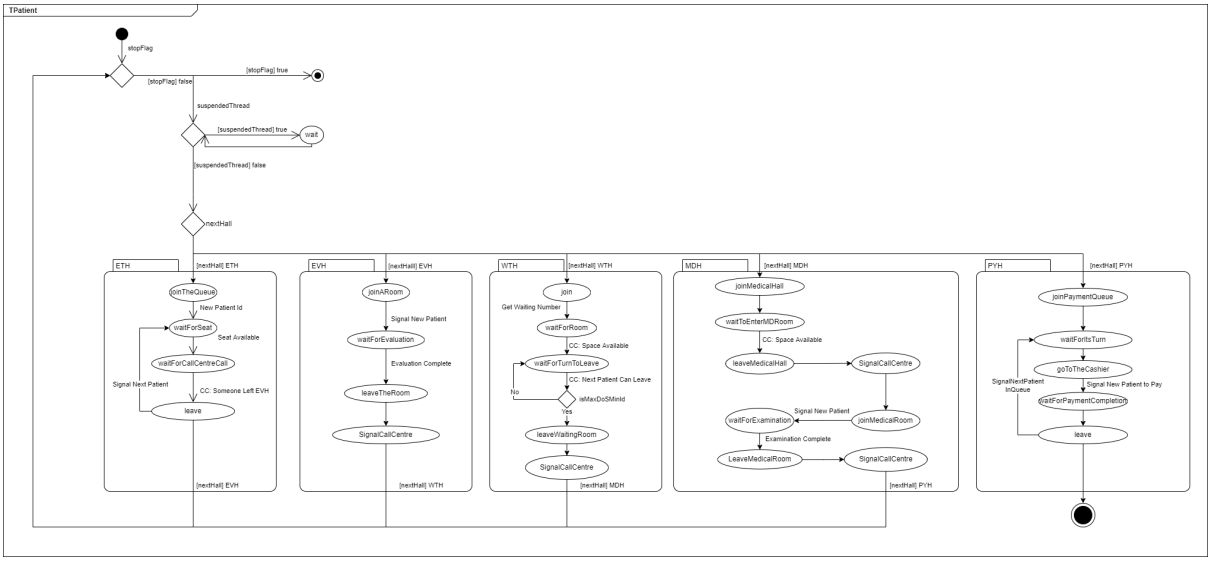
Diagrams

With our implementation, many classes and interfaces were developed in order to achieve our goal. Below is available a class diagram of our final implementations.



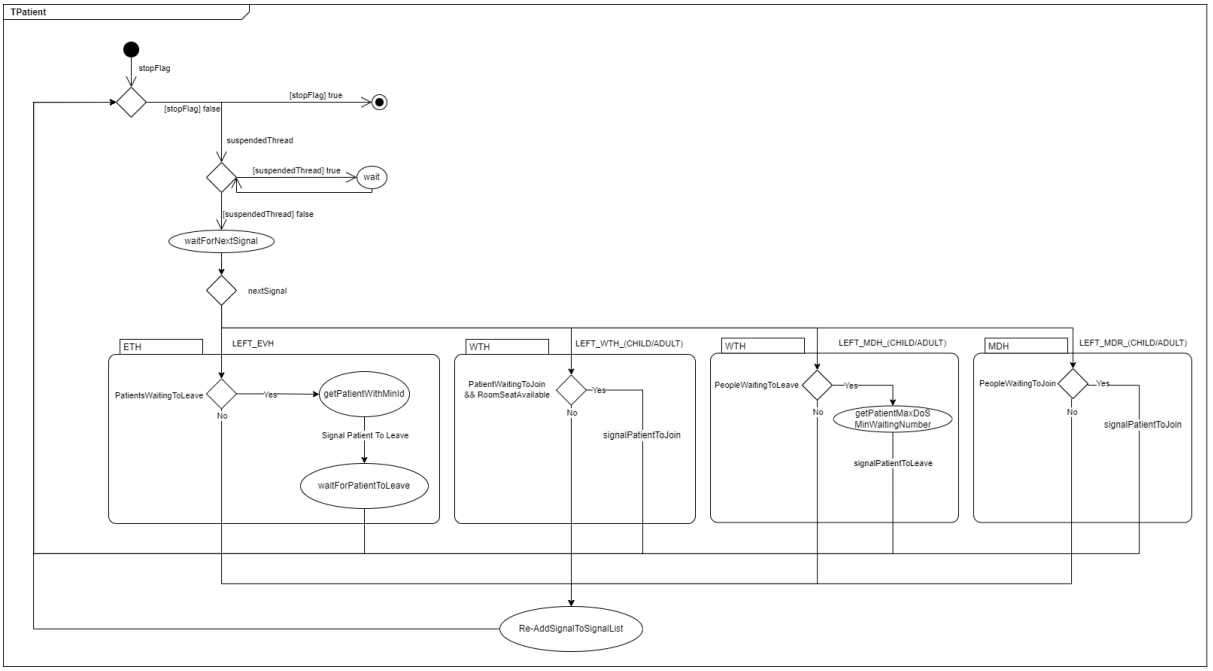
Patient

The Patient was the most complex thread entity of them all. As expected, the Patient moves through each Hall and has to wait for either a space available to sit or a Call Centre call to signal him to move. However, the process was similar to most of the Halls. The *'nextHall'* variable would define the hall the patient should try to enter next. As such it was initialized with the entrance Hall. The process that the Patient should follow is described below.



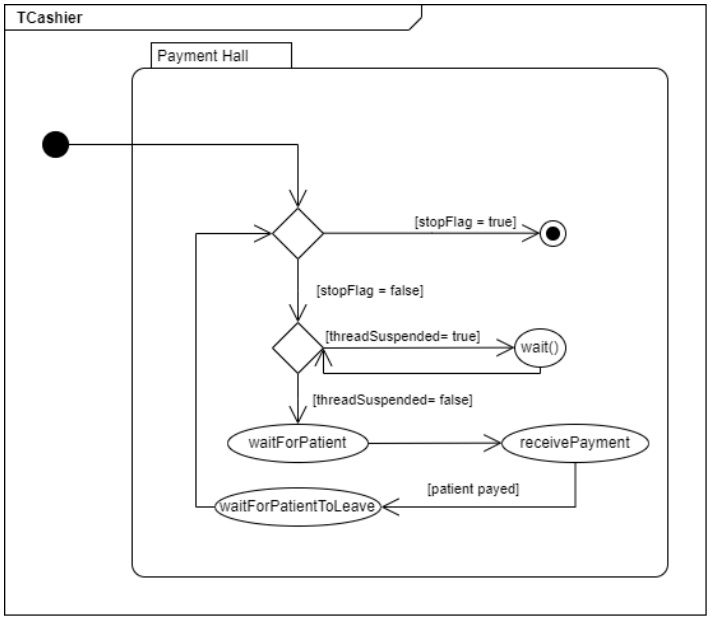
Call Centre

The Call Centre was responsible for allowing the movement of some patients. As expected this was possible because when a patient left a certain hall or a space was available, the call Centre signaled the next patient that it could move forward. If no patient was waiting to move, then the signal would be saved and added to the end of the list in order to not be lost and signal another patient in its due time, never stopping the operation. These signals are divided by the age of the patient, however, the process for the same category is the same.



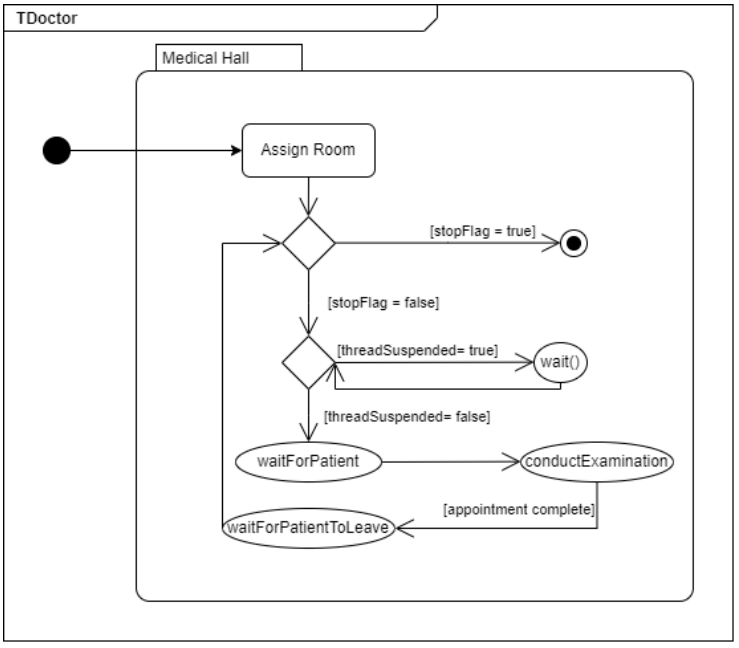
Cashier

The Cashier has the simplest of tasks. It simply waits for a patient and when there is one, proceeds with the payment. And keeps doing this until the simulation stops.



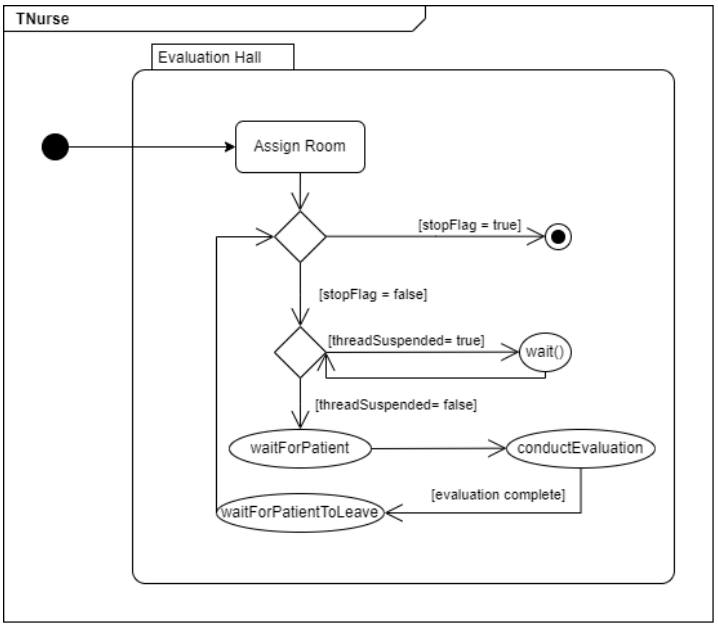
Doctor

The Doctor starts by requesting a room to be assigned. After this is completed, it simply keeps waiting in the room for a new patient to conduct the examination. After the examination is completed, it awaits for the same patient to leave to move on to the next patient.



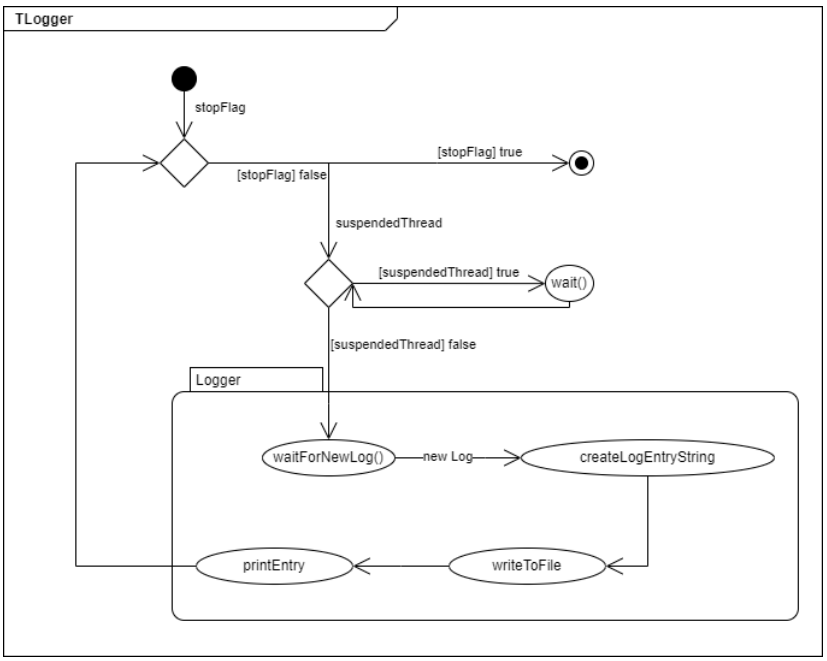
Nurse

The Nurse has the same operation as the Doctor. After being assigned a room, the nurse keeps waiting for a new patient to arrive and conduct an evaluation. After this process is finished, the Nurse waits for the patient to leave and evaluate the next one.



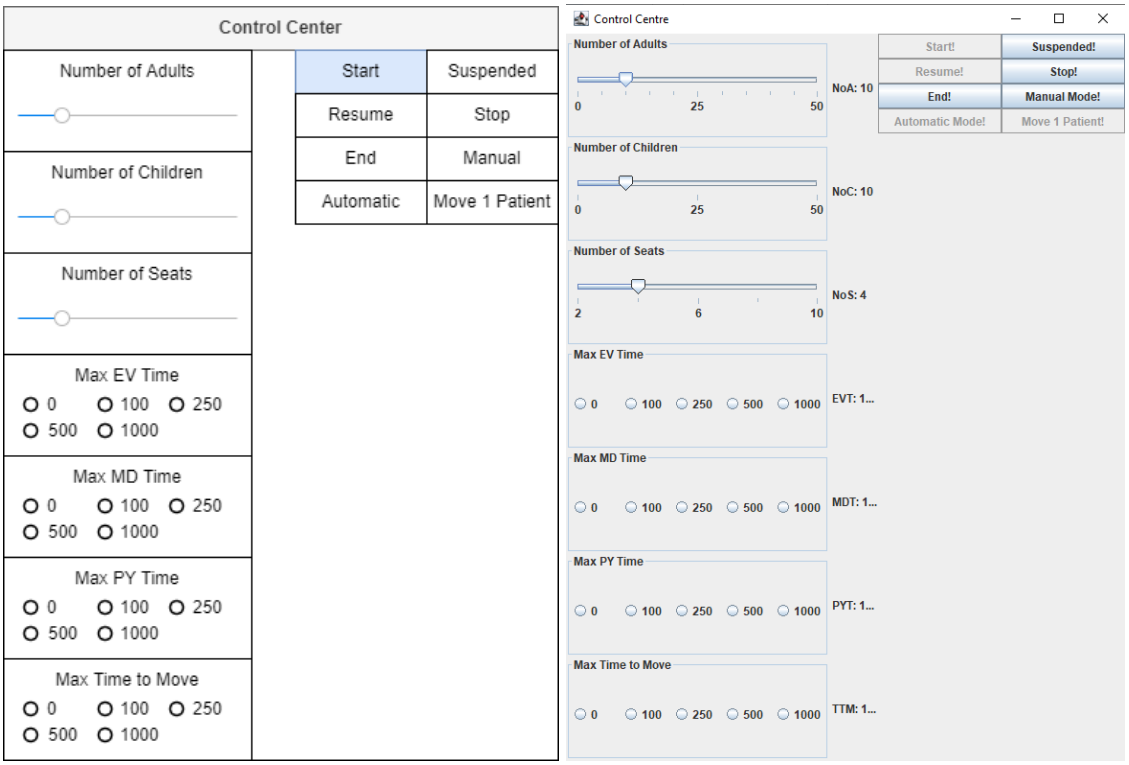
Logger

The Logger only stops when the connection is closed. It just keeps waiting for a new action to occur and then turns this into a log entry. This log entry is both printed in the console as well as written in the logs file inside the HCP.Logger package.



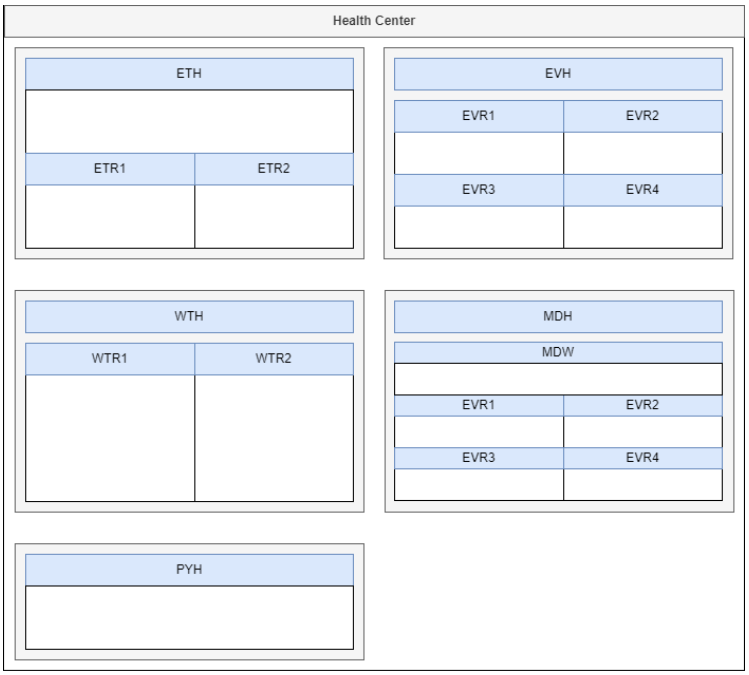
Graphical Interfaces

CCP GUI

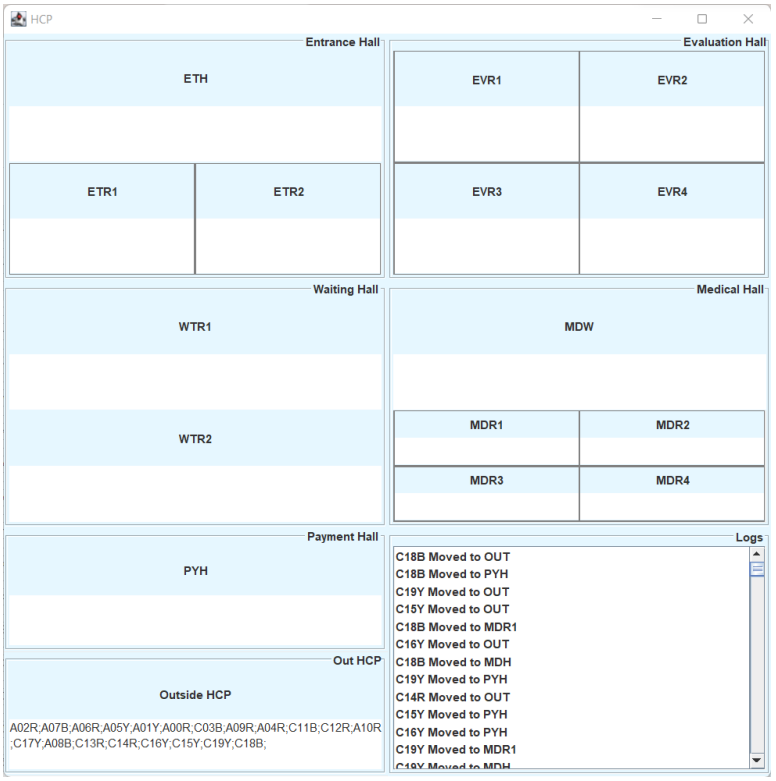


This layout includes all the seven parameters needed to be filled, with the number of adults, children, and seats with range input, and the times with radio boxes. These parameters already have a predefined value. The right of the layout has the existing running options. Our Implementation actually achieved a layout really similar to this one.

HCP GUI Layout



This interface displays the halls with their corresponding rooms, on the boxes below the hall/room name will appear the patient name as it moves through those spaces. Due to a lack of time, we were not able to further improve the appearance of our final implementation. However, we added a logging functionality that allows the user to better manage the changes in the simulation.



What was incorrectly implemented?

Manual Status

To use the manual status the *'Manual'* option in CCP has to be selected and then the *'Allow 1 Patient'* option is pressed to pass to the following state. However this is not the case, this button simply allows the Call Centre to process a single signal. As such, a specific **Patient X** is obliged to move. However, this can trigger another patient Y to move, who was not expecting a Call Centre call but simply the vacant seat which was previously occupied by **X**.

What has not been implemented?

Every requirement was implemented.

Contribution

Pedro Marques: 60%
Inês Leite: 40%