# Practical Assignment I (Health Centre)

## 1 Introduction

The practical assignment I is focused on an architecture where concurrency (processes and threads) is a key aspect.

## 2 Description

The project is about a simulation of the management process in a small health centre where the Manchester Emergency Triage System (METS) is implemented. The METS is a *clinical assessment and management tool for use in Emergency Departments and Ambulance Service face-to-face contact*. The METS has been adapted to address specific local requirements. It is based on an evaluation process carried out by nurses where each patient is assigned a degree of severity (DoS) before a medical appointment. Rules:

- Each DoS is assigned a colour: red, yellow and blue
- Descending priority from red to blue

The general description of the simulation is as follows:

A patient arrives to the health centre and proceeds to the Entrance Hall where he waits for his turn. Upon a call from the Call Centre, a patient is evaluated regarding its DoS and then he proceeds to the Waiting Hall. Upon a call from the Call Centre, the patient proceeds to the Medical Hall and waits for his turn. Finally, he pays his medical appointment.

The simulation comprises two main entities (java processes): Health Centre Process (HCP) and the Control Centre Process (CCP). The HCP is the entity responsible for the patients' treatments and the CCP is the entity responsible for controlling the simulation.

## 3 General Requirements and Constraints

- Two processes:
  - the CCP: where configuration, control and supervision takes place;
  - HCP: where patients' treatments are carried out
  - each process has its own GUI
- Technologies:
  - programming language: Java
  - Java Processes

- Java Threads
- Java Concurrency: implicit and explicit monitors
- SWING for GUI
- The communication between CCP and HCP is based on sockets and at the server side it must support multiple-clients simultaneously;
- It is not allowed to resort to standard java classes to implement any type of queues.

# 4 Health Centre Process

## 4.1 Halls

HCP comprises several halls and rooms.

**Entrance Hall (ETH)**

Hall where patients wait for the assessment of their DoS:

- The ETH comprises 2 rooms: $ETR_1$ for children and $ETR_2$ for adults
- Each $ETR_i$ has a defined number of seats (NoS/2)
- Patients can only enter the ETH if there are seats available in their corresponding $ETR_i$
- Upon entering the ETH, each patient is given a unique incremental sequential number (ETN) and then he proceeds to his corresponding $ETR_i$
- Patients leave the ETH in ascending order of their ETN and proceed to the EVH upon a call from the Call Centre.

**Evaluation Hall (EVH)**

Hall where the DoS is evaluated for each patient:

- EVH comprises 4 rooms: $EVR_1$, …, $EVR_4$
- In each $EVR_i$ there is one nurse
- Each EVR can assess both children and adults, one at a time
- Each patient is assigned a coloured bracelet corresponding to his DoS
- Each evaluation may take a variable period of time (EVT)
- Patients leave the EVH after the evaluation and proceed to the WTH

**Waiting Hall (WTH)**

The WTH is the first Hall where patients wait for their medical appointment:

- The WTH comprises two rooms: $WTR_1$ for children and $WTR_2$ for adults
- Each $WTR_i$ has a defined number of seats (NoS/2)
- Upon arriving at the WTH, each patient is given a unique incremental sequential number (WTN)
- In case there are no available seats, the patient must wait. Otherwise, he proceeds to his corresponding $WTR_i$

- Upon a call from the Call Centre, the patient with higher priority proceeds to the MDH
- Priority policy for patients' medical appointments (there is no distinction between adults and children):
    - Patients with higher DoS have higher priority
    - Patients with the same DoS are ordered by their WTN

**Medical Hall (MDH)**

Hall where medical appointments take place:

- The MDH comprises:
    - 1 waiting room (MDW)
    - 4 medical rooms: $MDR_1$, …, $MDR_4$
- MDW
    - The MDW is the second and final stage where patients wait for their medical appointment
    - The MDW has 2 seats: one for one child and one for one adult
    - Patients proceed to the medical appointment whenever the corresponding MDR is available
- $MDR_i$:
    - $MDR_1$ and $MDR_2$ are for children only and $MDR_3$ and $MDR_4$ are for adults only
    - Each $MDR_i$ has one doctor and one seat for one patient only at a time
    - Each medical appointment may take a variable period of time (MDT)
    - Patients leave the $MDR_i$ after the medical appointment and proceed to the payment hall

**Payment Hall (PYH)**

Hall where payments take place:

- The PYH comprises:
    - An entrance hall where patients can wait
    - A cashier where 1 patient pays his medical appointment
    - Patients pay by their order of arrival at the PYH (PYN)
    - After having payed the bill, patients leave the hospital
    - Each payment may take a variable period of time (PYT)

**Call Centre Hall (CCH)**

Hall where entity responsible for managing the movement of all Patients is:

- ETH -> EVH:
    - Whenever a Patient leaves a EVR he must inform the CCH
    - Whenever there is an empty EVR, the CLH must inform a Patient waiting in the ETH
- WTH -> $WTR_i$:

    - Whenever a Patient leaves a $WTR_i$, he must inform the CCH
    - Whenever there is an empty seat in a $WTR_i$, the CCH must inform a Patient waiting in the queue
- $WTR_i$ -> MDW
    - Whenever a Patient leaves a MDW, he must inform the CCH
    - Whenever there is an empty seat in a MDW, the CCH must inform a Patient waiting in the WTR
- MDW -> $MDR_i$
    - Whenever a Patient leaves a $MDR_i$, he must inform the CCH
    - Whenever there is an empty seat in a $MDR_i$, the CCH must inform a Patient waiting in the MDW

## 4.2 Operating Modes

The CCH has two operating modes:

- Auto: simulation runs automatically
- Manual: CCH can only authorize one movement for one Patient only at a time under command of the CCP
- The operating mode can change at any time, including when a simulation is running

## 4.3 Active Entities

The following active entities (Threads) must be implemented:

- Patient: entity willing a medical appointment (one Thread by Patient)
- Call Center: entity that partially supervises the movement of patients
- **Nurse: entity that evaluates patients regarding their DoS**
- Cashier: entity that accepts the payment

## 4.4 HCP Interfaces

The HCP must provide the following interfaces:

- Halls, Rooms and Seats
- Patients

# 5 Control Centre Process

The CCP GUI must implement the following requirements:

- Interfaces to be made available by the CCP GUI for the simulations configuration.
    - Number of adult (NoA) patients:
        - NoA = [1, 50]: n ∈ ℕ
        - Default value: 10
    - Number of children (NoC) patients:

        - NoC = [1, 50]: n ∈ ℕ
        - Default value: 10
    - NoS = {2, 4, 6, 8, 10}, default 4 (in each Hall)
    - EVT, 0 or a random number between 0 and {100, 250, 500, 1000} ms, default 100
    - MDT, 0 or a random number between 0 and {100, 250, 500, 1000} ms, default 100
    - PYT = 0 or a random number between 0 and {100, 250, 500, 1000} ms, default 100
    - Time to move (after waking-up) between halls or rooms for any patient (thread):
        - ttm = 0 or a random number between 0 and {100, 250, 500, 1000} ms, default 100
- Interfaces to be made available by the CCP GUI to control the simulation:
    - Start button:
        - to start a new simulation
        - Patients are created and go to the Entrance Hall
    - Suspend button:
        - to suspend the running simulation
        - Patients, Call Center and Cashier suspend their activity
    - Resume button:
        - to resume the running simulation
        - Patients, Call Centre and Cashier resume their normal activity
    - Stop button:
        - to stop the simulation
        - simulation evolves to its initial state and all Patients die
    - End button:
        - the two processes end
    - Operating mode:
        - option = {manual, auto}
        - button to authorize the CCH to allow the movement of one Patient (on manual mode only)

# 6 Simulation Logger

The state of the simulation must be printed in a text file (LOG.TXT) and also in the console. The logger must be implemented as Monitor. The simulation state is monitored at the HCP only. The format is as follows:

```
NoA:XX, NoC:YY, NoS: ZZ

STT | ETH ET1 ET2 | EVR1 EVR2 EVR3 EVR4 | WTH  WTR1 WTR2 | MDH  MDR1 MDR2 MDR3 MDR4 | PYH | OUT

XXX | YNN YNN YNN | YNNC YNNC YNNC YNNC | YNNC YNNC YNNC | YNNC YNNC YNNC YNNC YNNC | YNN | YNN
```

Where:

- STT: INI(initial), RUN(running), SUS(suspended), END(end), STO(stop), MAN(manual), AUT(automatic)
- OUT: patients have payed and left the Health Centre
- YNNC:
  - Y=A(adult) or C(child)
  - NN=ETN, WTN or PYN (in case NN is not updated, the last value must be kept)
  - C=R(red), Y(yellow), B(blue)

Example (partial simulation)

| STT | ETH | ET1 | ET2 | EVR1 | EVR2 | EVR3 | EVR4 | WTH | WTR1 | WTR2 | MDH | MDR1 | MDR2 | MDR3 | MDR4 | PYH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| INI | | | | | | | | | | | | | | | | |
| RUN | | | | | | | | | | | | | | | | |
| | A01 | | | | | | | | | | | | | | | |
| | | A01 | | | | | | | | | | | | | | |
| | A02 | | | | | | | | | | | | | | | |
| | C01 | | | | | | | | | | | | | | | |
| | | A02 | | | | | | | | | | | | | | |
| | | | | A01 | | | | | | | | | | | | |
| MAN | | | | | | | | | | | | | | | | |
| | | | C01 | | | | | | | | | | | | | |
| | | | | A01B | | | | | | | | | | | | |

# 7 Working Plan

- Improve specifications
- Design state diagrams, namely for active entities (Threads)
- Identify share areas (Monitors), their methods and their data structures
- Layouts design for CCP and HCP GUI;
- ….

## 8 Project Organization

You must follow the next guidelines:

- One NetBeans project only
- Two root Packages:
  - HC for all Health Centre Packages and files
  - CC for all Control Centre Packages and files
- Project name:
  - PA1CXGY
  - Where: X - > class number ∈ {1, 2}, Y -> group number
- All Thread Classes must start with the letter T, such as TPatient
- All Monitors must start with the letter M, such as MFifo
- Each Monitor must be a individual java class
- needed interface
- All Interfaces must start with the letter I, such as IProducer
- The access to Monitors must be through Interfaces provided their accessors
- Threads that access Monitors must receive as argument the correspondent interfaces
- All variables must be private (if not, a justification must be provided)
- Variables whose initial value does not change must be declared as final
- ReentrantLock is much more flexible than synchronized methods and provide additional functionalities. Use of ReentrantLock is mandatory. Nevertheless, you must know how synchronized methods work.
- Methods' and variables' names must be semantically oriented whenever convenient

## 9 Evaluation

The keys for the evaluation are:

- Requirements implementation
- Usability: user friendly GUI
- JavaDoc
- final report:
  - what has not been implemented and/or not correctly implemented
  - contribution of each student in %