Rapport du projet SOA

Automatic management of INSA's rooms

réalisé par : *Johan ALBERTI & Diep VU*

groupe : 5ISS_A2 - *Janvier-2020*

Github repository : https://github.com/hdvu95/SOA-project.git

Requirement specification

We were requested to develop a Web application (Proof-of-Concept) for managing INSA's rooms . This application must allow automatic closing windows, doors, turning on heating, turning off lights ... etc. This application relies on software services, sensors, and actuators. The goal is to retrieve data from sensors and analyze them to enable taking decisions. For instance, through software services, the application retrieves data of temperature and presence sensors of rooms and temperature sensors from outside, and for example depending on the values of the data, actions can be triggered. Our application must be based on a service-oriented architecture.

We decided the following scenarios for our autonomous management :

Scenario 1 :
A person enters room 1, the presence sensor detects it, the light switch on automatically.
Scenario 2 :
The temperature in Room 1 drops below (resp. above) 23°C, after checking the window status, the window is closed (resp. opened) automatically.
Scenario 3 :
The presence sensor detects someone in one of the rooms after 10 pm, after verification the alarm is activated.
Scenario 4 :
Scenario 2 + temperature adjustment with the radiator if necessary.
Scenario 5 :
After 10pm all room doors and windows are closed after checking the different presence sensors.

Project Management

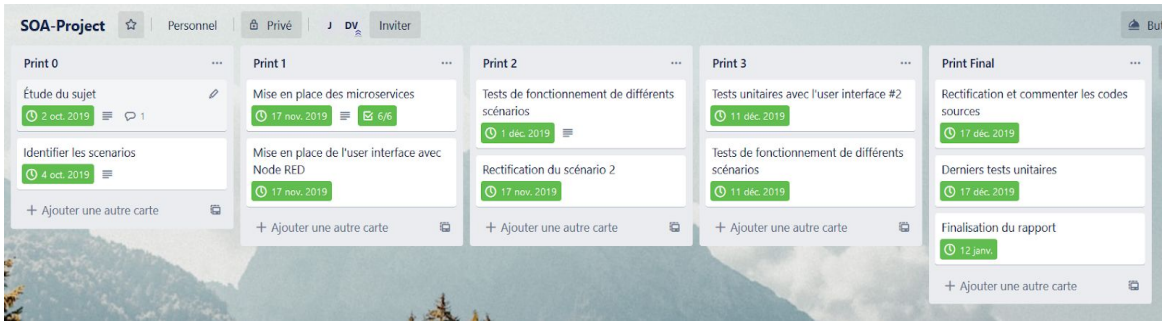

Figure 1: Scrum management method

As you can see we choose to use Scrum method to manage our project on Trello platform. All the prints are detailed on the website that you can find here.

Inside the project

The following figure describes our general architecture :

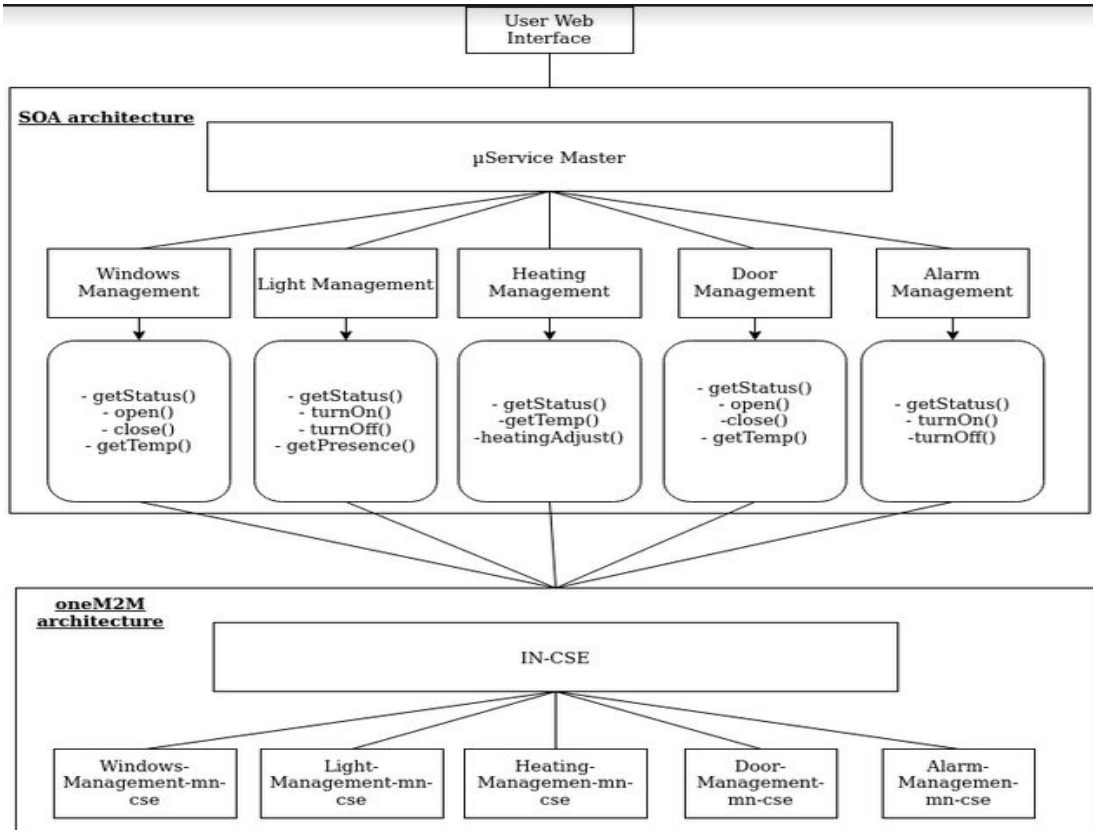

Figure 2: General architecture

Back-End

For this part, we have created one Springboot project for each microservice management with the displayed features in the figure2.

For the automatic aspect, in the ServiceMaster project, the *ServiceMasterApplication.java* class executes an infinite loop in order to manage all the scenarios we have mentioned in the introduction. (cf. Figure 3)

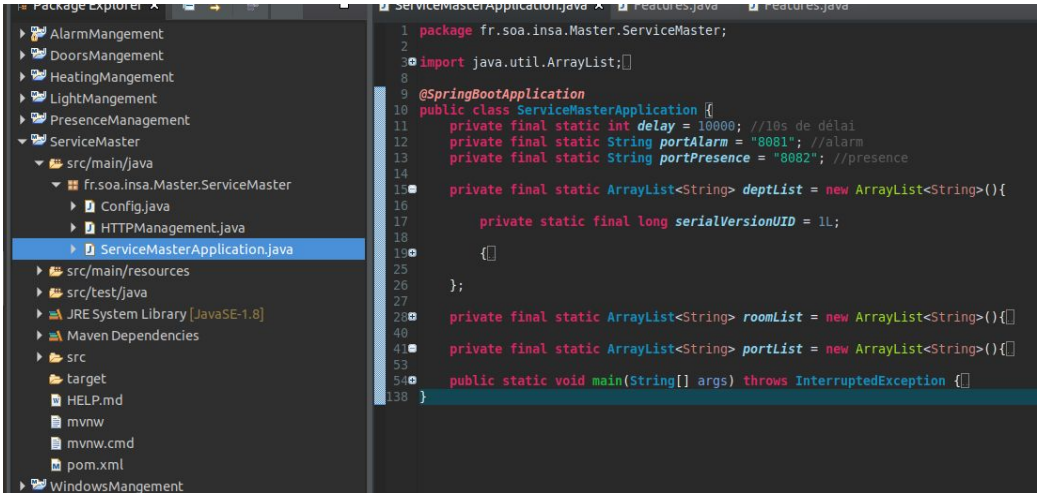

Figure 3: ServiceMasterApplication.java class

For the intelligent aspect, each project has its own *Features.java* class (cf. Figure 4) which describes what the microservice is able to offer as features, thanks to this class we are able to create an user interface with multiple functionalities (details in next part).

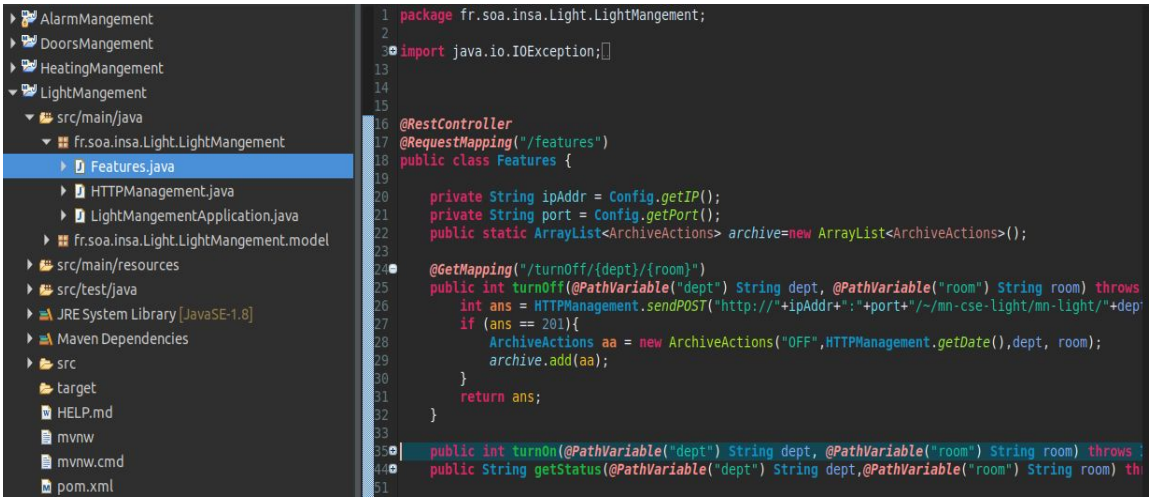

Figure 4: Features.java class

Front-End

In parallel with the back end development, we decided to implement a front end using Node-red, a powerful tool for IoT dashboard design.

This dashboard monitor INSA's rooms by sending and receiving HTTP request directly from all microservices created with Springboot.
The structure of the back-end has been designed to simplify HTTP request. Indeed, each monitored parameters (Light, Presence, Heating... see figure 2) correspond to a certain port.
Thus, with dropdown node the user set the room and the building that he want to check or manage.
Then, by concatenating the room and the building with the right port and function according to the user's choices, a request is sent to the corresponding microservice.
The functions used are :

getStatus()
tunOn()
turnOff()

Let's take the example of room GEI 103, where the user wishes to switch on the radiator at a temperature of 22°C. After clicking on Appliquer (to apply his choices) and Rafarîchir (to update the room status), the user is presented with the following screen:

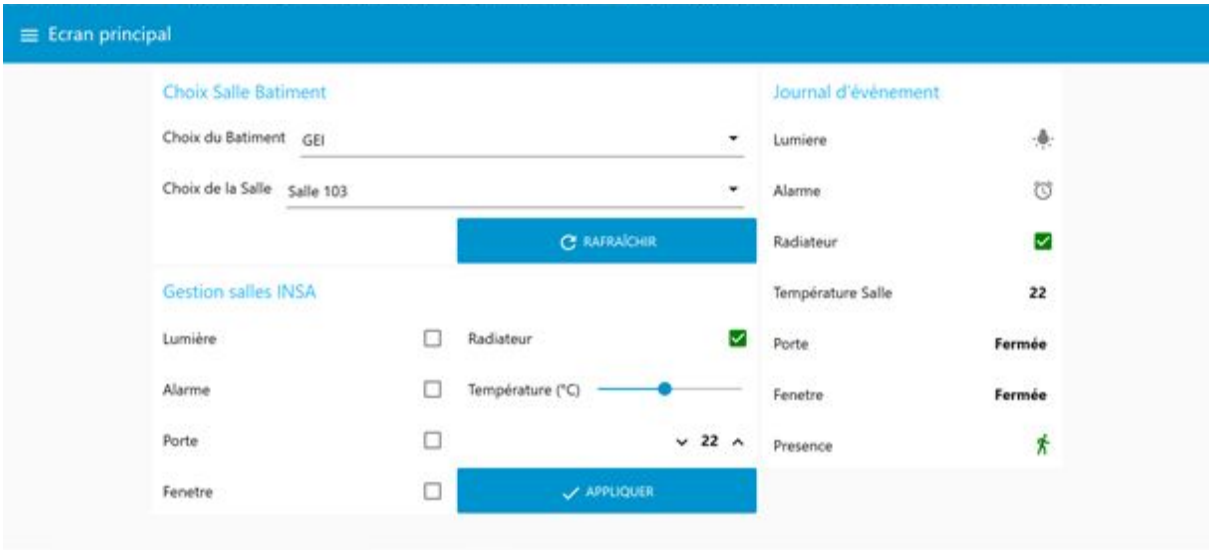

Figure 5: Dashboard

Now the user wants to turn on the light in the same room, open the door and turn on the radiator at the temperature of 27°C, this is what he gets :

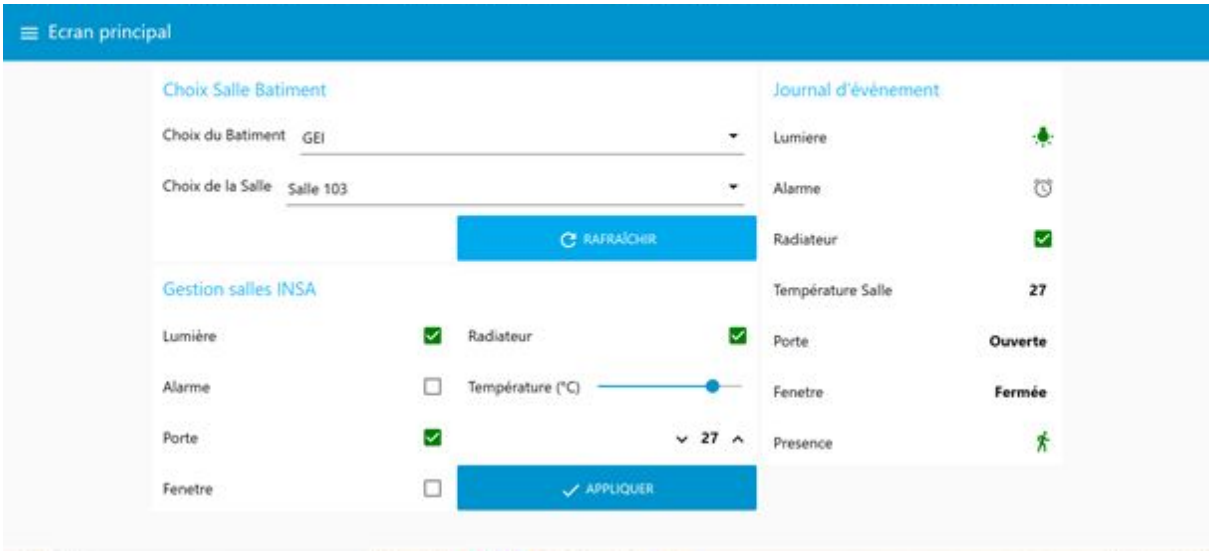

Figure 6: Dashboard use case

The dashboard also includes a history section that tracks all requests sent to the rooms which is downloadable as txt for the user in C:\Users\Public\Documents :

Figure 7: History

Conclusion

As a conclusion, this project has presented to us a global vision of Service Oriented Architecture development and we were able to improve our technical skills in this domain.

And we would like to thank Mme GUERMOUCHE Nawal and the persons in charge of the TPs for their precious advice.