INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

INNOVATIVE SMART SYSTEMS

INSA Toulouse

Innovative Smart Systems

Service Oriented Architecture

Alex NOIZE - Mathieu RAYNAUD - Group B2
Karima KHADIR

Promo 2019

January 24, 2019

Contents

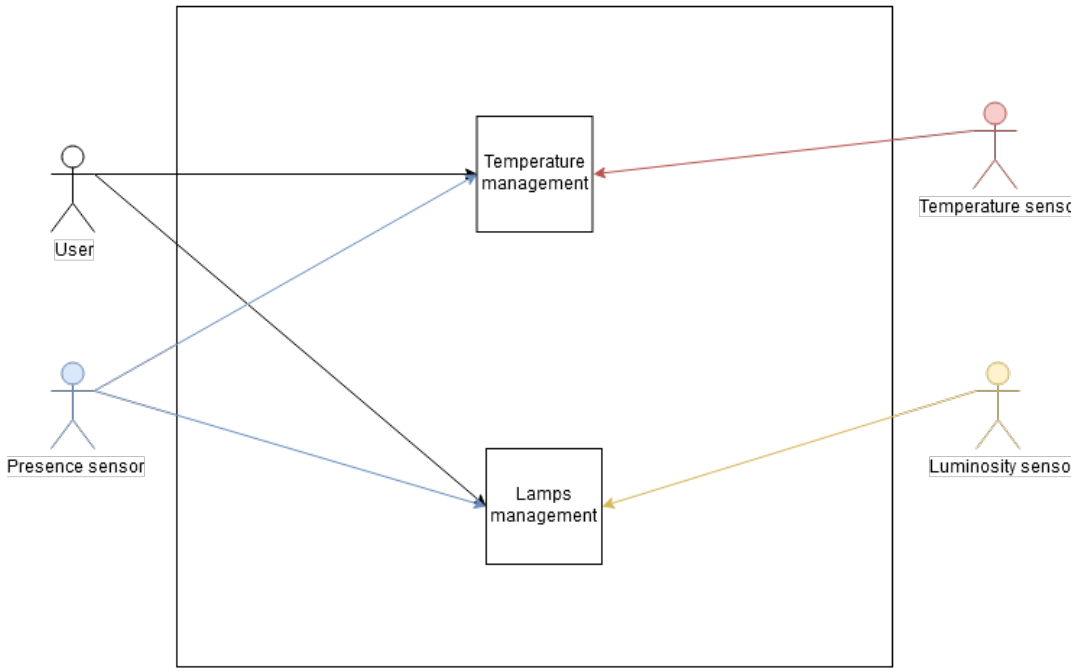
Chapter 1

Application design

For this project, we had to develop a smart room system based on a Rest API to control some parameters in INSA's class rooms. The system must be able through to control the elements of the rooms through a web interface and take decisions based on the data retrieved from the sensors.

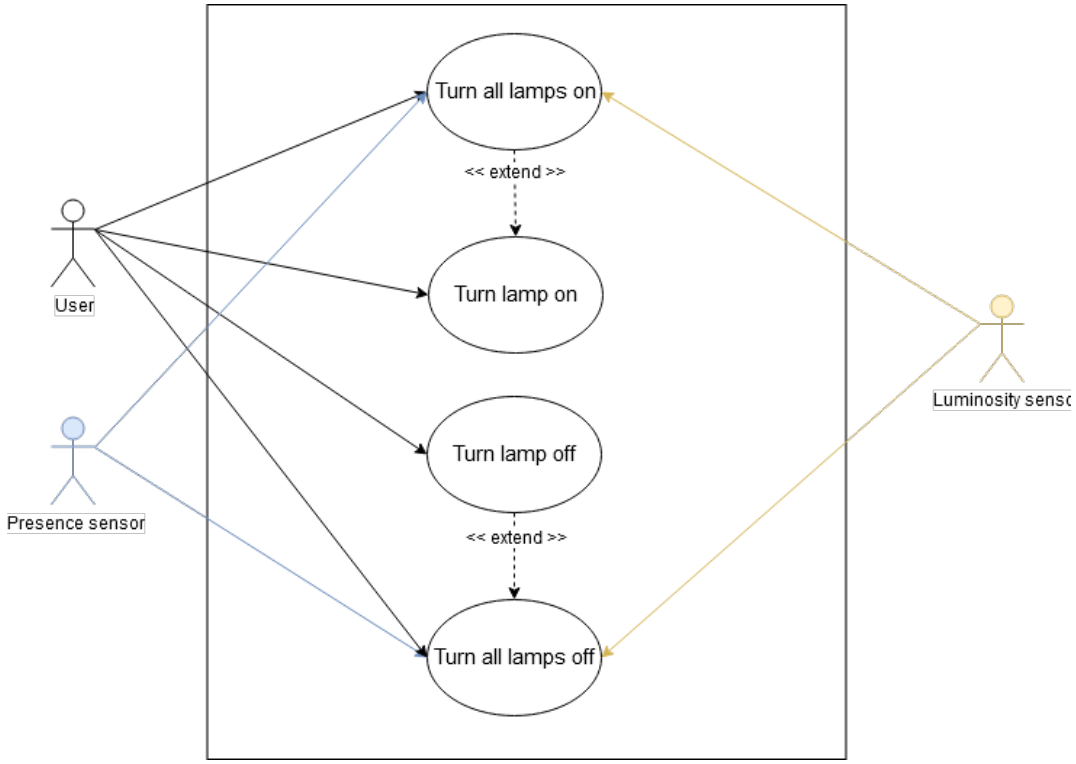
To give us a base line from where to start our development, we created the following use case diagrams :

1.1 global diagram



This diagram illustrates the global use cases of our application, which is composed of light related actions, and temperature related actions. Here are the use case for the lamps management :

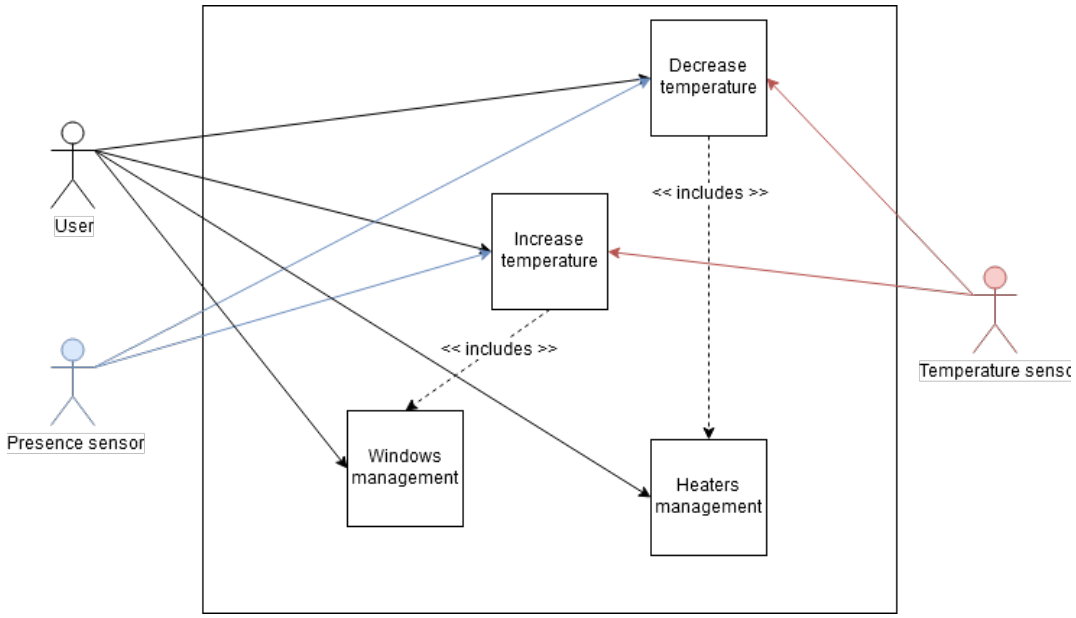
1.2 Lamps management



You can see that the management of the lamps is mostly based on the presence sensor.

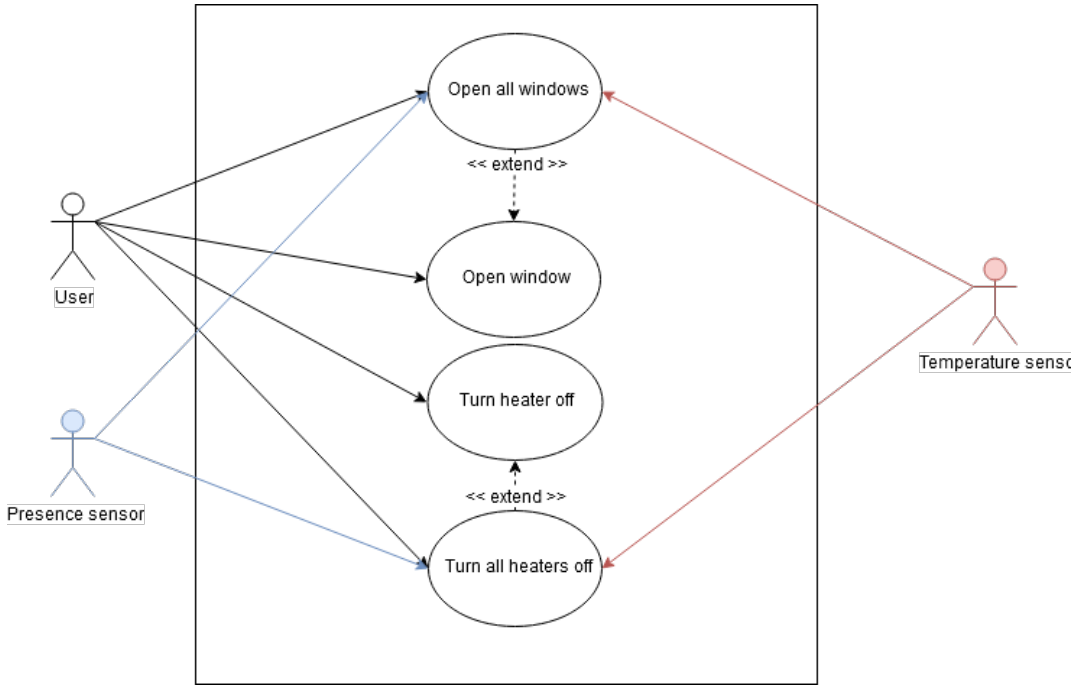
1.3 Temperature management

Here are the use case diagram for the global temperature management uses cases, followed by the detailed graphs for the different sub use cases:

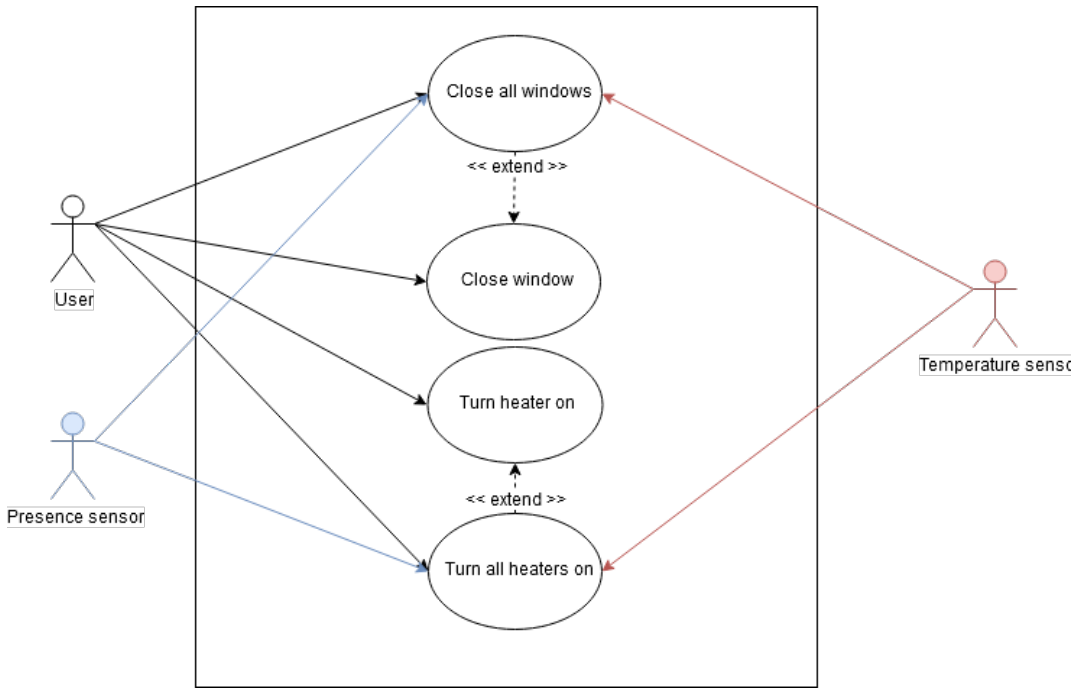


Lets's see now the details for the temperature management :

Decrease temperature :
https://www.overleaf.com/9221384352cmfyxqpxprsr

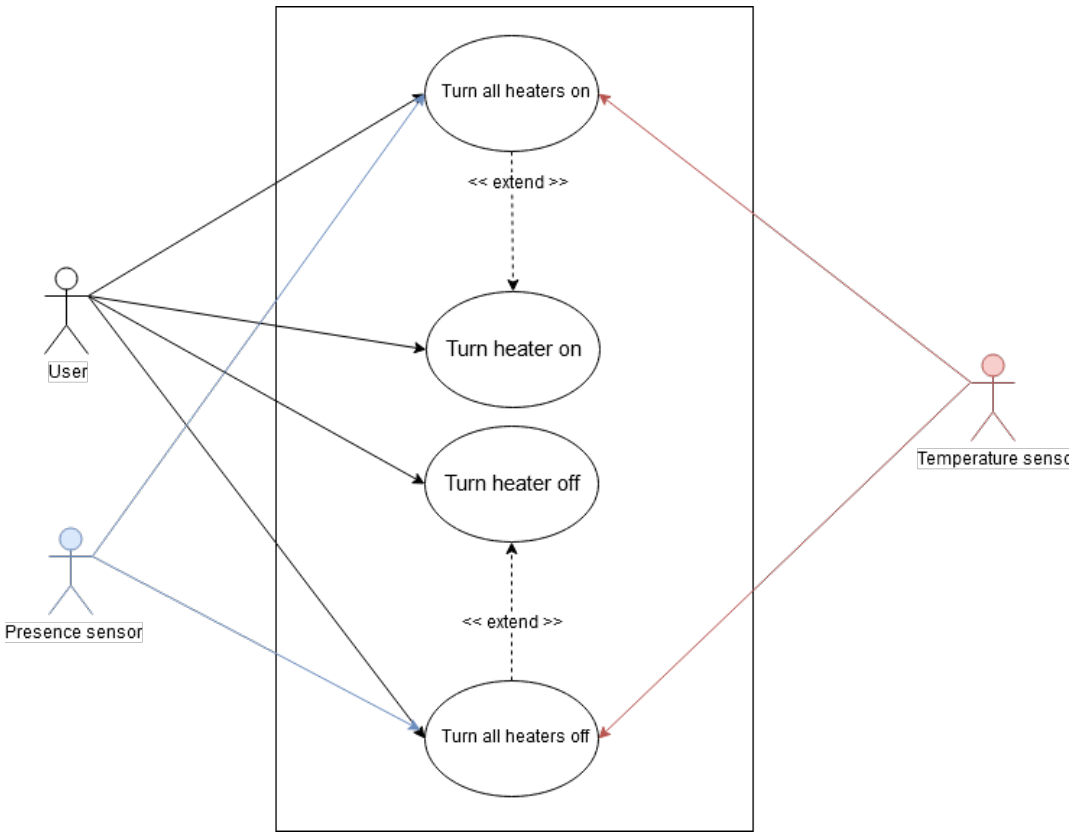


Increase temperature :

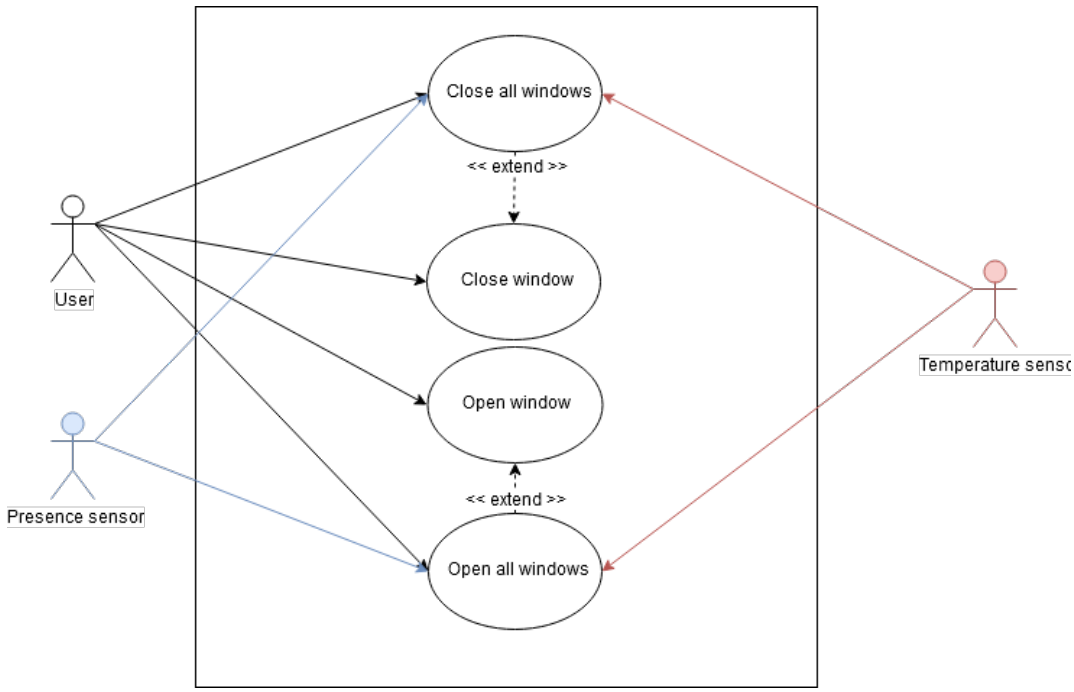


And finally the details for the management of the Heaters and the windows :

1.3.1 heater management



1.3.2 Window management



From there, we could focus on the architecture to chose for this web application and begin the Scrum method sprints to develop it.

Chapter 2

Architecture of our solution

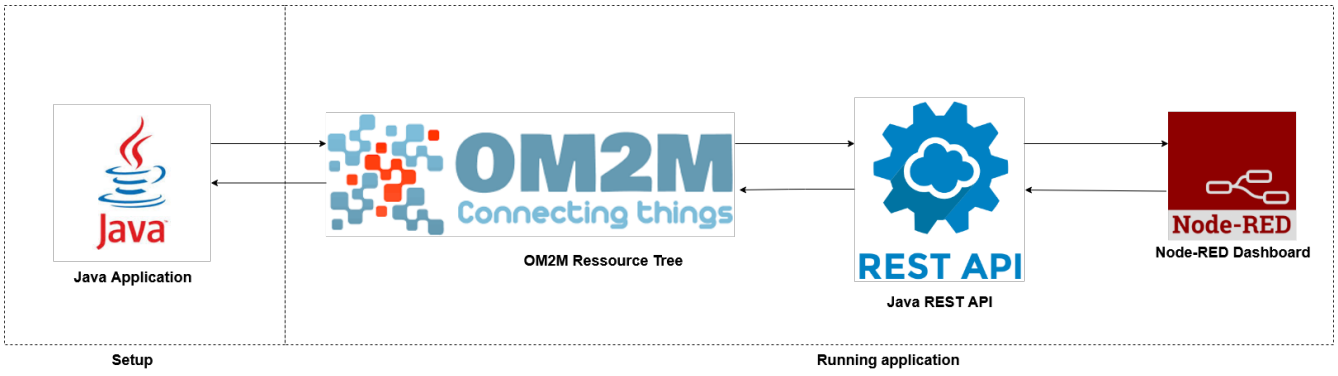
As specified, our architecture should be based on a Rest API to retrieve data from the sensors and give commands to the actuators.

To help us with that, we chose to combine this Restful application hosted on a Tomcat server with a OM2M solution and a node-RED server.
This way, we could easily simulate the devices with some OneM2M Application Entity hosted on the mn-cse gateway of OM2M. This is also the target of our Rest API to retrieve data via GET requests and send commands via POST requests.

Then, we set up a node-RED server that would play a dual role. First, we used the node-RED IDE to develop a simple dashboard that permitted us to see the data retrieved by the Rest API and send commands manually (shut the lights off, close the windows, turn the heater on ...).
On the other hand, the node-RED was handy to program the decision making, depending on the data from the sensors. This way we could program the use cases, for example turn on the heaters when the temperature is bellow a certain threshold.

Finally, the node-RED server enabled us to program scenarios to test our use cases. We could modify from there the state of the sensors via the Restful application and create various situations.

You will find below a chart of our architecture and a recap of the technology we used :



Chapter 3

Sprints and Features

For the development of this application, we used the Scrum Agile method and accomplished several sprints. Here are all the sprints, their goals and how they went

3.1 Sprint 1 : Set up of the architecture

Our first sprint after we specifying the architecture was to deploy our architecture. As we were a team of three, we each took one part of the architecture to deploy.
Here are the tasks for this sprints :

- Create the resource tree for the OM2M servers
- Deploy the Restful application on a Tomcat server and create the different path to retrieve data and send commands
- Deploy the node-RED server and proceed to some tests

During this two days sprint. We were able to accomplished all of the specified tasks and validate them through tests.

We first tested the link between the OM2M application and the Restful application. To do that we used postman to send GET and POST requests to the Tomcat server to see if the Rest API have indeed transmitted the requests to the OM2M Rest application. We could check that by monitoring the in-cse web interface and see if the POST requests were successful.

Next we tested the link between the Node-RED server and the Restful client, and by extension, to the OM2M servers. To do that we simply created POST and GET requests via the graphical programming interface of Node-RED, and deployed them to the server. We could visualize the GET requests through the debug nodes of Node-RED and the POST requests on the OM2M web interface.

All the tests were a success on time so we could proceed to next sprint.

3.2 Sprint 2 : Creation of the control dashboard

This next sprint was focused mainly on the Node-RED dashboard and was partially executed in parallel with the first sprint, up to the tests with the rest of the architecture.
This sprint was divided into two tasks :

- Designing the monitoring dashboard
- Designing the input dashboard

The role of the monitoring dashboard is to visualize the state and the value of the censors. It serves also as a control panel for basic commands, like switching the lights, turning on the heaters and opening the windows.

The input dashboard serves as a more advanced control panel, and is used for testing. We can for example set the current data for the temperature sensor at a certain number, and see if it triggers an event that we want to test.

After designing those panels and setting up the architecture, we had to test if we could retrieve the data from the OM2M server through the Restful client and display it on the dashboard. We were able to do so and even setting the value of the sensor through the input panel.

This two and an half day sprint was a success, so we moved on to the new sprint.

3.3 Sprint 3 : Light related use cases

After designing our application, setting up the architecture and the dashboard, we could begin implement the features. We first focused on the lights related use cases, as depicted in our use case diagram.
This 1/2 day sprint was pretty straight forward and was composed of one main task :

- Implement on a third Node-RED flow the use case Lamps management

The main challenge of this feature was implementing the conditions on a Node-RED flow. Getting the data from the motion sensor and setting the light was easy, as we already implemented that in the previous sprint. However, combining the different conditions dynamically with Node-RED was a bit tricky, but we eventually figured out how to create multi-conditional statements. When this was done, it was much easier to create complicated statements, and the last sprint went more smoothly.

3.4 Sprint 4 : Temperature related use cases

As this sprint was more complicated but using bricks that look a lot like the last sprint, it was important to itemize the tasks:

- Implement the heaters management
- Implement the windows management
- Implement the temperature increase
- Implement the temperature decrease

As we already created the algorithmic structure of such type of statement, this sprint took less time than expected. The 1/2 day sprint took us only roughly 1 hour.

These different sprints were quite short because the deadline of this project was soon after the beginning of the project. However, we identified quickly what the sprints were going to be and what was needed for each sprint. After the initial implementation of the infrastructure, the rest of the sprints were quicker to accomplish.

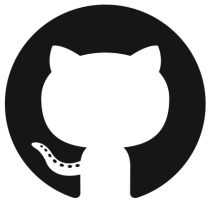
Conclusion

This project was a great opportunity to create a Restful application that can be implemented in a real scenario.

We were able to realize a Proof of Concept covering various use cases and that is very user friendly. From the base that we developed, the only thing missing from using it in real life is to set up real devices with our OM2M servers. Theoretically, we could use reuse the work we did during the Third and Fourth Practical Work of the middleware course to implement it with the HUE Lights and the motion sensor in minutes.

We managed our project thanks to the Scrum method and stayed in time throughout the duration of it. We realized 4 successful sprints and tested the features thoroughly.

At the end we were very happy with the results of this project.



https://github.com/AlexNoize/TP$_SOA.git$

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES TOULOUSE

INNOVATIVE SMART SYSTEMS

INSA Toulouse
135, avenue de Rangueil
31077 Toulouse Cedex 4 - France
www.insa-toulouse.fr

Université Fédérale Toulouse Midi-Pyrénées

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

MINISTÈRE
DE L'ÉDUCATION NATIONALE,
DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE